ADVANCE INFORMATION

82420EX PCISET DATA SHEET
82425EX PCI SYSTEM CONTROLLER (PSC) AND 82426EX ISA BRIDGE (IB)

- **Host CPU**
  - **— 25–33 MHz Intel486™ and OverDrive™ Processors**
  - **— L1 Write-Back Support**
- **Integrated DRAM Controller**
  - **— 1 to 128 MByte Main Memory**
  - **— 70 ns Fast Page Mode DRAM SIMMs Supported**
  - **— Supports 256 KByte, 1 MByte, and 4 MByte Double and Single Sided SIMMs**
  - **— Read Page Hit Timing of 3-2-2-2 at 33 MHz**
  - **— Burst Mode PCI Master Accesses**
  - **— Decoupled Refresh Reduces DRAM Latency**
  - **— Five RAS Lines**
- **Integrated L2 Cache Controller**
  - **— Write-Back and Write-Through Cache Policies**
  - **— Direct Mapped Organization**
  - **— 64, 128, 256 or 512 KByte Cache Sizes**
  - **— Programmable Zero Wait-State L2 Cache Read and Write Accesses**
  - **— Two Banks Interleaved or a Single Bank Non-Interleaved Operation**
  - **— No VALID Bit Required**
- **25/33 MHz PCI Bus Interface**
  - **— Two Bus Masters**
  - **— PCI Auto Configuration Support**
- **Host/PCI Bridge**
  - **— Converts Back-to-Back Sequential Memory Writes to PCI Burst Writes**
  - **— CPU Memory Write Posting to PCI**
- **PCI Local Bus IDE Interface**
  - **— Supports Mode 3 Timing**
- **Programmable Attribute Map for First 1 MByte of Main Memory**
- **100% ISA Compatible**
  - **— Directly Drives 5 ISA Slots**
- **Two 8237 DMA Controllers**
  - **— 7 DMA Channels**
  - **— 27-bit Addressability**
  - **— Compatible DMA Transfers**
- **One 82C54 Timer/Counter**
  - **— System Timer**
  - **— Refresh Request**
  - **— Speaker Tone**
- **Two 82C59 Interrupt Controllers**
  - **— 14 Interrupts**
  - **— Edge/Level Sense is Programmable per Channel**
  - **— PCI Interrupt Steering for Plug and Play Compatibility**
- **X-Bus Peripheral Support**
  - **— RTC, KBC, BIOS Chip Selects**
  - **— Control for Lower X-Bus Transceiver**
  - **— Integrates Mouse Interrupt**
  - **— Coprocessor Error Reporting**
- **Non-Maskable Interrupts (NMI)**
  - **— PCI System Errors**
  - **— Main Memory Parity Errors**
  - **— ISA Parity Errors**
- **System Power Management (Intel SMM Support)**
  - **— Programmable System Management Interrupt (SMI)—Hardware Events, Software Events, EXTSMI#**
  - **— Programmable CPU Clock Control**
  - **— Fast On/Off Mode**
- **Generates System Clocks**
- **160-Pin QFP Package for IB**
- **208-Pin QFP Package for PSC**

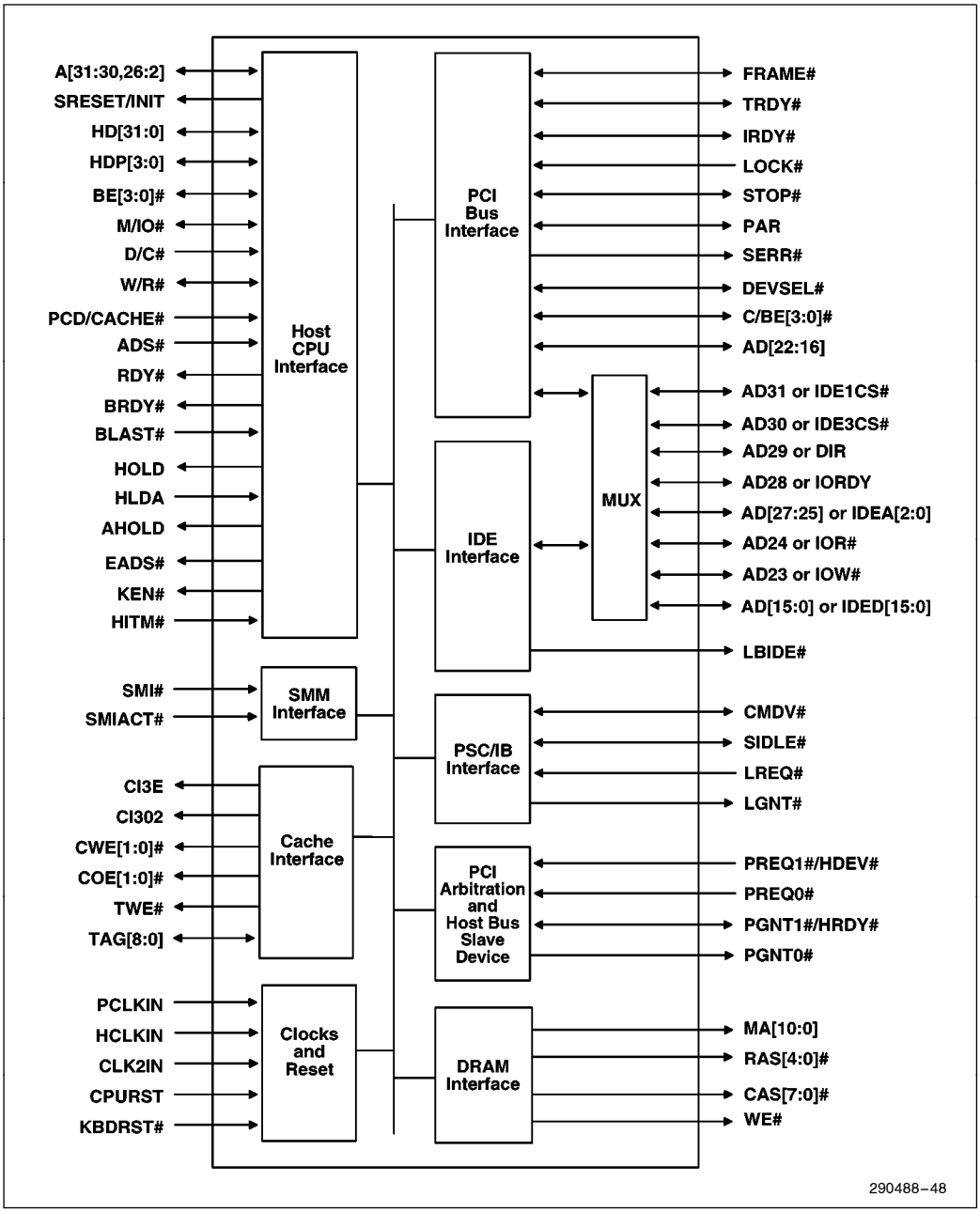
December 1995

Order Number: 290488-004

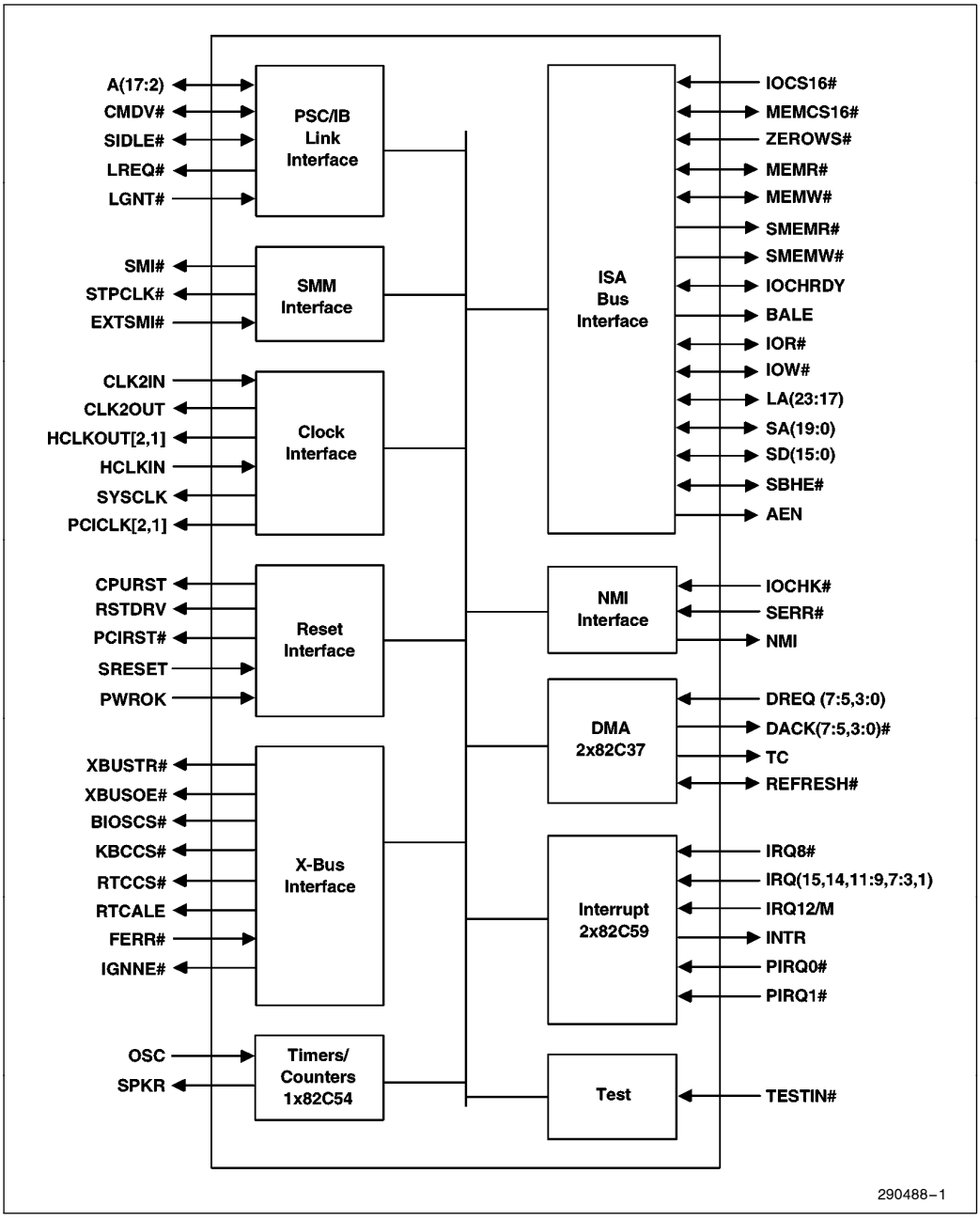
The 82420EX PCIset is the foundation for the **Value Flexible Motherboard** solution for entry-level Intel486™ processor-based PCI systems. The Value Flexible Motherboard solution, including 82420EX, Intel486 processor, 82091AA Advanced Integrated Peripherals, 82C42 Keyboard Controller, Flash BIOS, and Plug & Play software, drives PCI into the mainstream. The 82420EX PCIset is a highly integrated solution enabling low cost, small form factor motherboard designs. All Intel486 processors and upgrades are supported, including L1 write-back and Intel SMM power management. PCI Local Bus IDE is incorporated for higher performance IDE at no additional cost.

The 82420EX was designed from the ground up for PCI performance. It consists of two components—the 82425EX PCI System Controller (PSC) and the 82426EX ISA Bridge (IB). The PSC integrates the L2 cache controller and the DRAM controller. The cache controller supports both write-through and write-back cache policies and cache sizes from 64 KBytes to 512 KBytes in an interleaved or non-interleaved configuration. The DRAM controller interfaces main memory to the Host Bus and the PCI Bus. The PSC supports a two-way interleaved DRAM organization for optimum performance. Up to ten single-sided SIMMs or four double-sided and two single-sided SIMMs provide a maximum of 128 MBytes of main memory. The PSC provides memory write posting to PCI for enhanced CPU-to-PCI memory write performance. In addition, the PSC provides a high performance PCI Local Bus IDE interface.

The IB is the bridge between the ISA Bus and Host Bus, and integrates the common I/O functions found in today's ISA-based PC systems—a seven channel DMA controller, two 82C59 interrupt controllers, an 8254 timer/counter, Intel SMM power management support, and control logic for NMI generation. The IB also provides the decode for external BIOS, real time clock, and keyboard controller. Edge/Level interrupts and interrupt steering are supported for PCI plug and play compatibility. The IB integrates the ISA address and data path, reducing TTL and system cost. In addition, the integration of system clock generation logic eliminates the need for external host and PCI clock drivers.



82425EX PCI System Controller (PSC) Block Diagram



82426EX ISA Bridge (IB) Block Diagram

82420EX PCISET DATA SHEET
82425EX PCI SYSTEM CONTROLLER (PSC)
AND 82426EX ISA BRIDGE (IB)

CONTENTS

PAGE

1.0 PINOUT INFORMATION 11
1.1 PSC Pin Assignment 12
1.2 IB Pin Assignment 17

2.0 SIGNAL DESCRIPTION 22
2.1 PSC Signals 22
2.1.1 HOST CPU INTERFACE SIGNALS (PSC) 22
2.1.2 SECONDARY CACHE SIGNALS (PSC) 24
2.1.3 PCI SIGNALS (PSC) 25
2.1.4 SYSTEM POWER MANAGEMENT (SMM) SIGNALS (PSC) 25
2.1.5 DRAM CONTROL SIGNALS (PSC) 26
2.1.6 PSC/IB LINK INTERFACE (PSC) 26
2.1.7 PCI BUS ARBITRATION/HOST BUS SLAVE DEVICE (PSC) 26
2.1.8 PCI BUS IDE (PSC) 27
2.1.9 CLOCKS AND RESET (PSC) 28
2.2 IB Signals 28
2.2.1 ISA INTERFACE SIGNALS (IB) 28
2.2.2 NMI SIGNALS (IB) 30
2.2.3 DMA SIGNALS (IB) 30
2.2.4 TIMER/COUNTER SIGNALS (IB) 31
2.2.5 INTERRUPT CONTROLLER SIGNALS (IB) 31
2.2.6 X-BUS SIGNALS (IB) 32
2.2.7 PSC/IB LINK INTERFACE SIGNALS (IB) 33
2.2.8 SYSTEM POWER MANAGEMENT (SMM) SIGNALS (IB) 33
2.2.9 SYSTEM CLOCK SIGNALS (IB) 34
2.2.10 SYSTEM RESET SIGNALS (IB) 34
2.2.11 TEST SIGNALS (IB) 34

3.0 REGISTER DESCRIPTION 35
3.1 Register Access 35
3.2 I/O Control Registers 42
3.2.1 CONFADD—CONFIGURATION ADDRESS REGISTER 42
3.2.2 CONFDATA—CONFIGURATION DATA REGISTER 42
3.2.3 TRC—TURBO/RESET CONTROL REGISTER 43

CONTENTS

PAGE

3.3 PCI Configuration Registers 44
3.3.1 VID—VENDOR IDENTIFICATION REGISTER 44
3.3.2 DID—DEVICE IDENTIFICATION REGISTER 44
3.3.3 PCICOM—PCI COMMAND REGISTER 44
3.3.4 DS—DEVICE STATUS REGISTER 45
3.3.5 RID—REVISION IDENTIFICATION REGISTER 46
3.3.6 PCICON—PCI CONTROL REGISTER 46
3.3.7 HOSTDEV—HOST DEVICE CONTROL REGISTER 48
3.3.8 LBIDE—PCI LOCAL BUS IDE CONTROL REGISTER 48
3.3.9 IORT—ISA I/O RECOVERY TIMER REGISTER 51
3.3.10 PREV—PART REVISION REGISTER 52
3.3.11 XBCSA—X-BUS CHIP SELECT A REGISTER 52
3.3.12 HOSTSEL—HOST SELECT REGISTER 54
3.3.13 DFC—DETURBO FREQUENCY CONTROL REGISTER 54
3.3.14 SCC—SECONDARY (L2) CACHE CONTROL REGISTER 55
3.3.14.1 L2 Write Timing 56
3.3.15 DRAMC—DRAM CONTROL REGISTER 57
3.3.16 PAM[6:0]—PROGRAMMABLE ATTRIBUTE MAP REGISTERS 58
3.3.17 DRB—DRAM ROW BOUNDARY REGISTERS 60
3.3.18 PIRQ1RC/PIRQ0RC—PIRQ ROUTE CONTROL REGISTERS 61
3.3.19 DMH—DRAM MEMORY HOLE REGISTER 62
3.3.20 TOM—TOP OF MEMORY 62
3.3.21 SMRAMCON—SMRAM CONTROL REGISTER 63
3.3.22 SMICNTL—SMI CONTROL REGISTER 65
3.3.23 SMIEN—SMI ENABLE REGISTER 66
3.3.24 SEE—SYSTEM EVENT ENABLE 67
3.3.25 FTMR—FAST OFF TIMER REGISTER 68
3.3.26 SMIREQ—SMI REQUEST REGISTER 69
3.3.27 CTLTMRL—CLOCK THROTTLE STPCLK# LOW TIMER 70
3.3.28 CTLTMRH—CLOCK THROTTLE STPCLK# HIGH TIMER 70
3.4 ISA-Compatible Registers 70
3.4.1 DMA REGISTER DESCRIPTION 70
3.4.1.1 DCOM—DMA Command Register 71
3.4.1.2 DCM—DMA Channel Mode Register 72
3.4.1.3 DREQ—DMA Request Register 73
3.4.1.4 WSMB—Write Single Mask Bit Register 73

CONTENTS

PAGE

3.4.1.5 WAMB—Write All Mask Bits Register 74
3.4.1.6 DS—DMA Status Register 75
3.4.1.7 DB&CA—DMA Base And Current Address Registers (8237 Compatible Segment) 75
3.4.1.8 DB&CBW—DMA Base And Current Byte/Word Count Registers (8237 Compatible Segment) 76
3.4.1.9 DMLPG—DMA Memory Low Page Registers 77
3.4.1.10 DMHPG—DMA Memory High Page Register 77
3.4.1.11 DCLBP—DMA Clear Byte Pointer Register 78
3.4.1.12 DMCL—DMA Master Clear Register 78
3.4.1.13 DCLM—DMA Clear Mask Register 78
3.4.2 TIMER/COUNTER REGISTER DESCRIPTION 79
3.4.2.1 TCW—Timer Control Word Register 79
3.4.2.2 TMSTAT—Interval Timer Status Byte Format Register 83
3.4.2.3 CAPS—Counter Access Ports Register 84
3.4.3 INTERRUPT CONTROLLER REGISTER DESCRIPTION 84
3.4.3.1 ICW1—Initialization Command Word 1 Register 84
3.4.3.2 ICW2—Initialization Command Word 2 Register 85
3.4.3.3 ICW3—Initialization Command Word 3 Register 86
3.4.3.4 ICW3—Initialization Command Word 3 Register 86
3.4.3.5 ICW4—Initialization Command Word 4 Register 87
3.4.3.6 OCW1—Operational Control Word 1 Register 88
3.4.3.7 OCW2—Operational Control Word 2 Register 88
3.4.3.8 OCW3—Operational Control Word 3 Register 90
3.4.3.9 ELCR1—Edge/Level Triggered Register 91
3.4.3.10 ELCR2—Edge/Level Triggered Register 91
3.4.4 X-BUS REGISTER DESCRIPTION 91
3.4.4.1 RIRQ—Reset IRQ[12,1] 92
3.4.4.2 CPERR—Coprocessor Error Register 92
3.4.5 NMI REGISTER DESCRIPTION 92
3.4.5.1 NMISC—NMI Status and Control Register 93
3.4.5.2 NMIERTC—NMI Enable and Real Time Clock Address Register 94
3.4.6 POWER MANAGEMENT REGISTER DESCRIPTION 94
3.4.6.1 APMC—Advanced Power Management Control Port 94
3.4.6.2 APMS—Advanced Power Management Status Port 95

CONTENTS

PAGE

4.0 FUNCTIONAL DESCRIPTION 95
4.1 Memory and I/O Address Map 95
4.1.1 MEMORY ADDRESS MAP 95
4.1.2 BIOS MEMORY SPACE 97
4.1.3 VIDEO FRAME BUFFER 97
4.1.4 I/O ACCESSES 97
4.1.5 SMRAM: PROTECTED SMM MEMORY BLOCK 97
4.2 PSC/IB Link Interface 98
4.3 Host CPU Interface 98
4.3.1 HOST BUS SLAVE DEVICE 98
4.3.2 L1 CACHE SUPPORT 99
4.3.3 SOFT AND HARD RESETS 99
4.3.4 KEYBOARD CONTROLLER (A20) 101
4.4 PCI Interface 101
4.4.1 PCI BUS CYCLES SUPPORT 102
4.4.2 HOST TO PCI CYCLES 103
4.4.3 PCI CYCLE TERMINATIONS 103
4.4.4 EXCLUSIVE CYCLES 104
4.4.5 PARITY SUPPORT 104
4.5 PCI Local Bus IDE 104
4.6 ISA Interface 106
4.7 X-Bus 107
4.7.1 COPROCESSOR ERROR FUNCTION 108
4.7.2 MOUSE FUNCTION 108
4.8 System Arbitration 108
4.8.1 SYSTEM ARBITRATION SCHEME 108
4.9 DMA Controller 109
4.10 Interval Timer 110
4.11 Interrupt Controller 110
4.11.1 PROGRAMMING THE INTERRUPT CONTROLLER 111
4.11.2 EDGE AND LEVEL INTERRUPT TRIGGERED MODE 111
4.11.3 INTERRUPT STEERING 112
4.12 L2 Cache 112
4.12.1 L2 CACHE SIZES/PERFORMANCE 112
4.12.2 CACHE OPERATIONS 113

CONTENTS

PAGE

4.12.3 CACHE CONSISTENCY 113
4.12.4 INITIALIZING THE L2 CACHE 114
4.12.5 CACHE LINE DESCRIPTION 114
4.12.6 L2 CACHE STRUCTURE 114
4.13 DRAM Interface 115
4.13.1 DRAM ADDRESS TRANSLATION 116
4.13.2 DRAM STRUCTURE 116
4.13.3 DRAM SIMM SIZE/DENSITY OPTIONS 118
4.13.4 DRAM PAGE MODE 118
4.13.5 PROGRAMMABLE WAIT-STATES 119
4.13.5.1 RAS Precharge 119
4.13.5.2 CAS Read Time 119
4.13.5.3 CAS Write Time 119
4.13.5.4 MA Setup Time 119
4.13.5.5 MA Hold Time 119
4.13.6 DRAM PERFORMANCE 119
4.14 Power Management 120
4.14.1 SMM MODE 121
4.14.2 SMI SOURCES 121
4.14.3 SMI (SMI#) AND INTERRUPT (INTR) ORDERING 122
4.14.4 CLOCK CONTROL 123
4.15 Clocks 123
4.15.1 CLOCK LAYOUT/LOADING RECOMMENDATION 124

5.0 DESIGN CONSIDERATIONS 125

6.0 ELECTRICAL CHARACTERISTICS 126
6.1 Maximum Ratings 126
6.2 PSC and IB DC Characteristics 127
6.3 IB AC Characteristics 129
6.3.1 CLOCK/RESET TIMINGS 129
6.3.2 PSC/IB LINK INTERFACE TIMINGS 130
6.3.3 SYSTEM POWER MANAGEMENT TIMINGS 130
6.3.4 ISA BUS AND X-BUS TIMINGS 131
6.3.5 AC TEST LOADS 140
6.3.6 AC TIMING WAVEFORMS 140

CONTENTS

PAGE

6.4 PSC AC Characteristics 157
6.4.1 HOST CLOCK TIMING 157
6.4.2 CPU INTERFACE TIMINGS 157
6.4.3 SECOND LEVEL CACHE TIMING 159
6.4.4 DRAM INTERFACE TIMING 159
6.4.5 PCI TIMING 160
6.4.6 PSC/IB LINK INTERFACE TIMING 161
6.4.7 PCI BUS IDE TIMING 161
6.4.8 AC TEST LOADS 161
6.4.9 MISCELLANEOUS CLOCK TIMINGS 161

7.0 IB AND PSC PACKAGE INFORMATION 164
7.1 Thermal Characteristics 165

8.0 TESTABILITY 166
8.1 PSC Testability 166
8.1.1 PSC TRI-STATE CONTROL 166
8.1.2 PSC NAND TREE 166
8.1.3 PSC NAND TREE DIAGRAM 173
8.2 IB Testability 174
8.2.1 IB TRI-STATE 174
8.2.2 IB NAND TREE 174
8.2.3 IB NAND TREE CELL ORDER 174
8.2.4 IB NAND TREE DIAGRAM 180

9.0 REVISION HISTORY 180

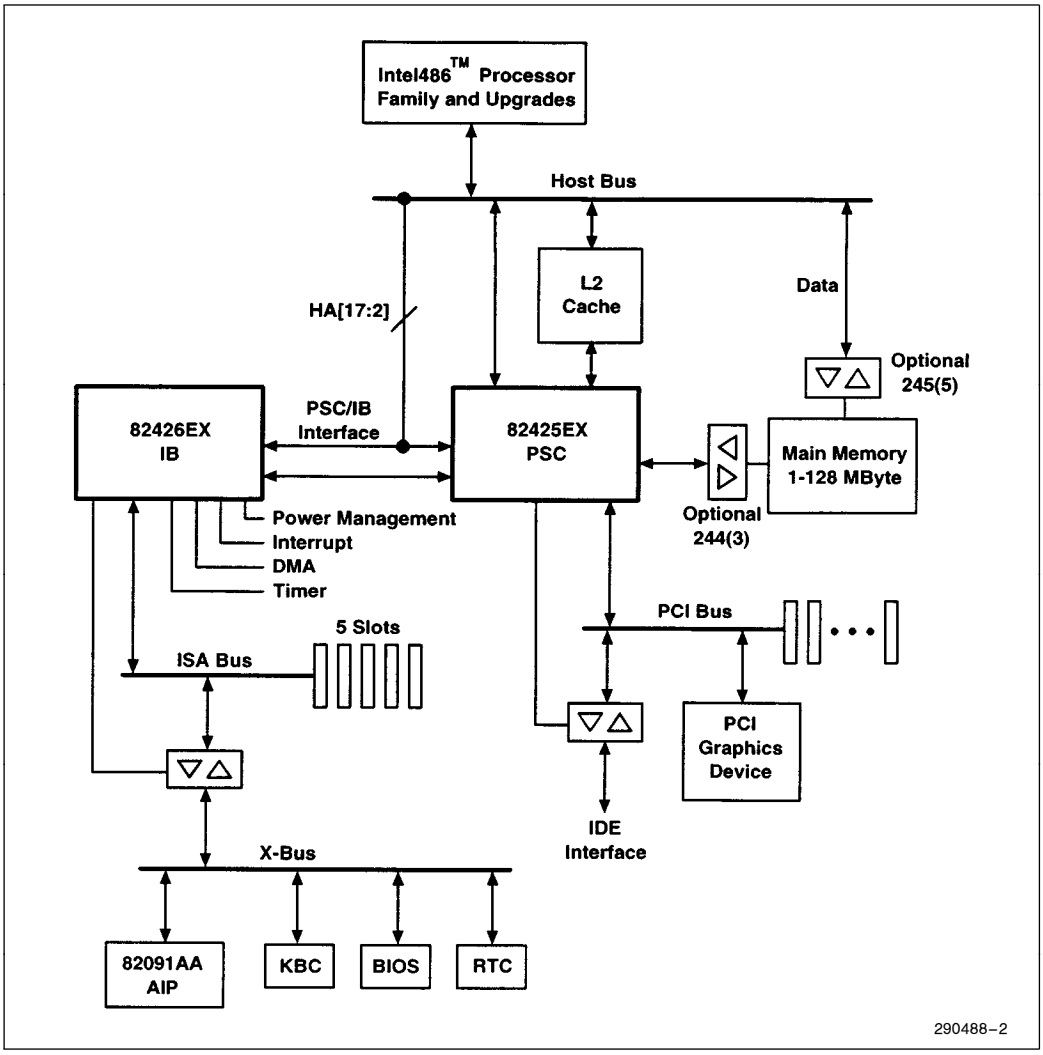


Figure 1. Example System Block Diagram

1.0 PINOUT INFORMATION

This section provides the PSC and IB pin assignment and package information. For each device, the pin assignments are listed in both alphabetical and numerical order.

1.1 PSC Pin Assignment

The PSC package is a 208-pin Quad Flatpack (QFP). Figure 2 shows the pin assignment on the package. Tables 1 and 2 list the pin assignments alphabetically and numerically, respectively.

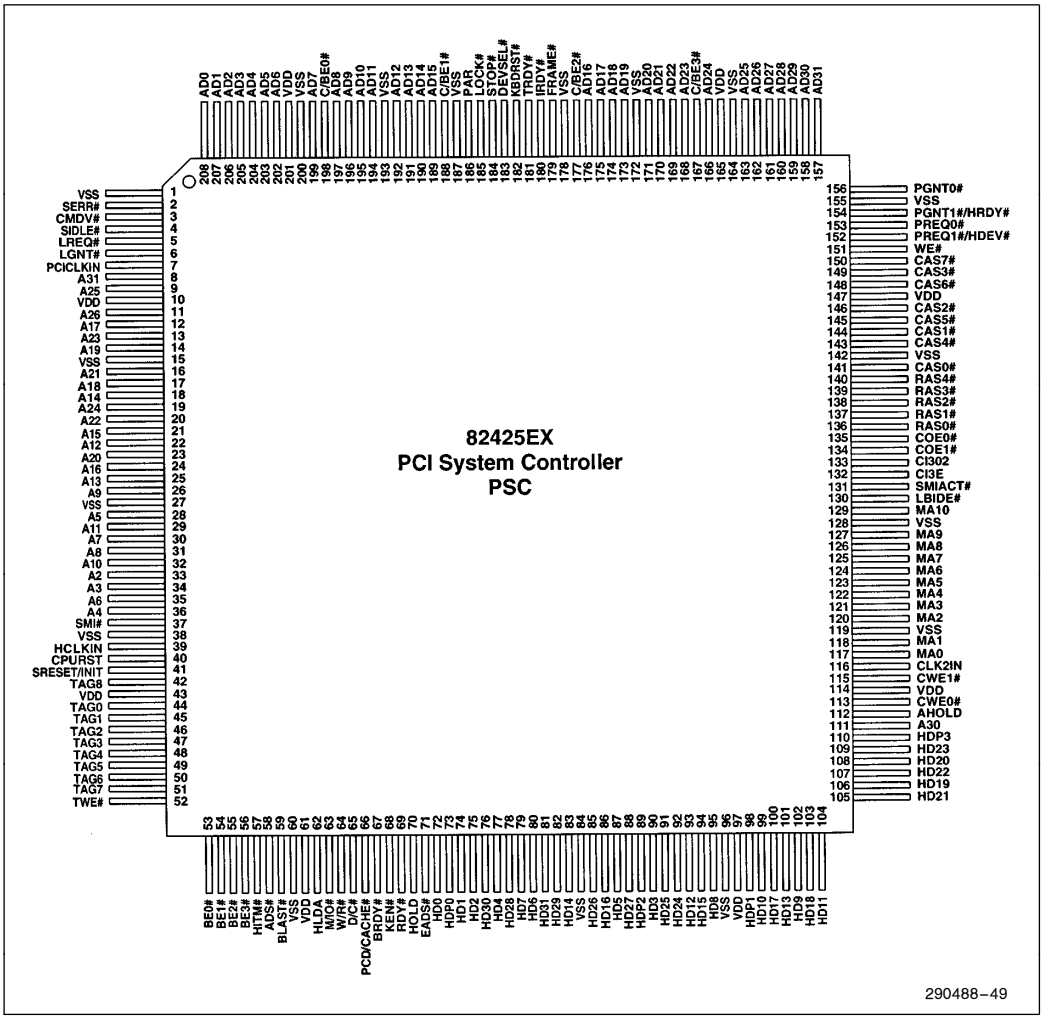


Figure 2. PSC Pin Assignment

Table 1. Alphabetical PSC Pin Assignment List

Pin Name	Pin #	I/O
A2	33	I/O
A3	34	I/O
A4	36	I/O
A5	28	I/O
A6	35	I/O
A7	30	I/O
A8	31	I/O
A9	26	I/O
A10	32	I/O
A11	29	I/O
A12	22	I/O
A13	25	I/O
A14	18	I/O
A15	21	I/O
A16	24	I/O
A17	12	I/O
A18	17	I/O
A19	14	I/O
A20	23	I/O
A21	16	I/O
A22	20	I/O
A23	13	I/O
A24	19	I/O
A25	9	I/O
A26	11	I/O
A30	111	I/O
A31	8	I/O
AD0	208	I/O
AD1	207	I/O
AD2	206	I/O
AD3	205	I/O
AD4	204	I/O

Pin Name	Pin #	I/O
AD5	203	I/O
AD6	202	I/O
AD7	199	I/O
AD8	197	I/O
AD9	196	I/O
AD10	195	I/O
AD11	194	I/O
AD12	192	I/O
AD13	191	I/O
AD14	190	I/O
AD15	189	I/O
AD16	176	I/O
AD17	175	I/O
AD18	174	I/O
AD19	173	I/O
AD20	171	I/O
AD21	170	I/O
AD22	169	I/O
AD23	168	I/O
AD24	166	I/O
AD25	163	I/O
AD26	162	I/O
AD27	161	I/O
AD28	160	I/O
AD29	159	I/O
AD30	158	I/O
AD31	157	I/O
ADS#	58	I
AHOLD	112	O
BE0#	53	I/O
BE1#	54	I/O
BE2#	55	I/O

Pin Name	Pin #	I/O
BE3#	56	I/O
BLAST#	59	I
BRDY#	67	O
C/BE0#	198	I/O
C/BE1#	188	I/O
C/BE2#	177	I/O
C/BE3#	167	I/O
CAS0#	141	O
CAS1#	144	O
CAS2#	146	O
CAS3#	149	O
CAS4#	143	O
CAS5#	145	O
CAS6#	148	O
CAS7#	150	O
CI3E	132	O
CI3O2	133	O
CLK2IN	116	I
CMDV#	3	I/O
COE0#	135	O
COE1#	134	O
CPURST	40	I
CWE0#	113	O
CWE1#	115	O
D/C#	65	I
DEVSEL#	183	s/t/s
EADS#	71	O
FRAME#	179	s/t/s
HCLKIN	39	I
HD0	72	I/O
HD1	74	I/O
HD2	75	I/O

Pin Name	Pin #	I/O
HD3	90	I/O
HD4	77	I/O
HD5	87	I/O
HD6	80	I/O
HD7	79	I/O
HD8	95	I/O
HD9	102	I/O
HD10	99	I/O
HD11	104	I/O
HD12	93	I/O
HD13	101	I/O
HD14	83	I/O
HD15	94	I/O
HD16	86	I/O
HD17	100	I/O
HD18	103	I/O
HD19	106	I/O
HD20	108	I/O
HD21	105	I/O
HD22	107	I/O
HD23	109	I/O
HD24	92	I/O
HD25	91	I/O
HD26	85	I/O
HD27	88	I/O
HD28	78	I/O
HD29	82	I/O
HD30	76	I/O
HD31	81	I/O
HDP0	73	I/O
HDP1	98	I/O
HDP2	89	I/O

Table 1. Alphabetical PSC Pin Assignment List (Continued)

Pin Name	Pin #	I/O
HDP3	110	I/O
HITM#	57	I
HLDA	62	I
HOLD	70	O
IRDY#	180	s/t/s
KBDRST#	182	I
KEN#	68	O
LBIDE#	130	O
LGNT#	6	O
LOCK#	185	I
LREQ#	5	I
M/IO#	63	I/O
MA0	117	O
MA1	118	O
MA2	120	O
MA3	121	O
MA4	122	O
MA5	123	O
MA6	124	O
MA7	125	O
MA8	126	O

Pin Name	Pin #	I/O
MA9	127	O
MA10	129	O
PAR	186	I/O
PCD/ CACHE	66	I
PCICLKIN	7	I
PGNT0#	156	O
PGNT1# / HRDY#	154	I/O
PREQ0#	153	I
PREQ1# / HDEV#	152	I
RAS0#	136	O
RAS1#	137	O
RAS2#	138	O
RAS3#	139	O
RAS4#	140	O
RDY#	69	O
SERR#	2	OD
SIDLE#	4	I/O
SMI#	37	I

Pin Name	Pin #	I/O
SMIACT#	131	I
SRESET/ INIT	41	O
STOP#	184	s/t/s
TAG0	44	I/O
TAG1	45	I/O
TAG2	46	I/O
TAG3	47	I/O
TAG4	48	I/O
TAG5	49	I/O
TAG6	50	I/O
TAG7	51	I/O
TAG8	42	I/O
TRDY#	181	s/t/s
TWE#	52	O
V_{DD}	10	V
V_{DD}	43	V
V_{DD}	61	V
V_{DD}	97	V
V_{DD}	147	V
V_{DD}	114	V

Pin Name	Pin #	I/O
V_{DD}	165	V
V_{DD}	201	V
V_{SS}	1	V
V_{SS}	15	V
V_{SS}	27	V
V_{SS}	38	V
V_{SS}	60	V
V_{SS}	84	V
V_{SS}	96	V
V_{SS}	119	V
V_{SS}	128	V
V_{SS}	142	V
V_{SS}	155	V
V_{SS}	164	V
V_{SS}	172	V
V_{SS}	178	V
V_{SS}	187	V
V_{SS}	193	V
V_{SS}	200	V
WE#	151	O
W/R#	64	I/O

Table 2. Numerical PSC Pin Assignment List

Pin Name	Pin #	I/O
V_{SS}	1	V
SERR#	2	OD
CMDV#	3	I/O
SIDLE#	4	I/O
LREQ#	5	I
LGNT#	6	O
PCICLKIN	7	I
A31	8	I/O
A25	9	I/O
V_{DD}	10	V
A26	11	I/O
A17	12	I/O
A23	13	I/O
A19	14	I/O
V_{SS}	15	V
A21	16	I/O
A18	17	I/O
A14	18	I/O
A24	19	I/O
A22	20	I/O
A15	21	I/O
A12	22	I/O
A20	23	I/O
A16	24	I/O
A13	25	I/O
A9	26	I/O
V_{SS}	27	V
A5	28	I/O
A11	29	I/O
A7	30	I/O

Pin Name	Pin #	I/O
A8	31	I/O
A10	32	I/O
A2	33	I/O
A3	34	I/O
A6	35	I/O
A4	36	I/O
SMI#	37	I
V_{SS}	38	V
HCLKIN	39	I
CPURST	40	I
SRESET/ INIT	41	O
TAG8	42	I/O
V_{DD}	43	V
TAG0	44	I/O
TAG1	45	I/O
TAG2	46	I/O
TAG3	47	I/O
TAG4	48	I/O
TAG5	49	I/O
TAG6	50	I/O
TAG7	51	I/O
TWE#	52	O
BE0#	53	I/O
BE1#	54	I/O
BE2#	55	I/O
BE3#	56	I/O
HITM#	57	I
ADS#	58	I
BLAST#	59	I
V_{SS}	60	V

Pin Name	Pin #	I/O
V_{DD}	61	V
HLDA	62	I
M/IO#	63	I/O
W/R#	64	I/O
D/C#	65	I
PCD/ CACHE#	66	I
BRDY#	67	O
KEN#	68	O
RDY#	69	O
HOLD	70	O
EADS#	71	O
HD0	72	I/O
HDP0	73	I/O
HD1	74	I/O
HD2	75	I/O
HD30	76	I/O
HD4	77	I/O
HD28	78	I/O
HD7	79	I/O
HD6	80	I/O
HD31	81	I/O
HD29	82	I/O
HD14	83	I/O
V_{SS}	84	V
HD26	85	I/O
HD16	86	I/O
HD5	87	I/O
HD27	88	I/O
HDP2	89	I/O
HD3	90	I/O

Pin Name	Pin #	I/O
HD25	91	I/O
HD24	92	I/O
HD12	93	I/O
HD15	94	I/O
HD8	95	I/O
V_{SS}	96	V
V_{DD}	97	V
HDP1	98	I/O
HD10	99	I/O
HD17	100	I/O
HD13	101	I/O
HD9	102	I/O
HD18	103	I/O
HD11	104	I/O
HD21	105	I/O
HD19	106	I/O
HD22	107	I/O
HD20	108	I/O
HD23	109	I/O
HDP3	110	I/O
A30	111	I/O
AHOLD	112	O
CWE0#	113	O
V_{DD}	114	V
CWE1#	115	O
CLK2IN	116	I
MA0	117	O
MA1	118	O
V_{SS}	119	V
MA2	120	O

Table 2. Numerical PSC Pin Assignment List (Continued)

Pin Name	Pin #	I/O
MA3	121	O
MA4	122	O
MA5	123	O
MA6	124	O
MA7	125	O
MA8	126	O
MA9	127	O
V_{SS}	128	V
MA10	129	O
LBIDE#	130	O
SMIACT#	131	I
C13E	132	O
C13O2	133	O
COE1#	134	O
COE0#	135	O
RAS0#	136	O
RAS1#	137	O
RAS2#	138	O
RAS3#	139	O
RAS4#	140	O
CAS0#	141	O
V_{SS}	142	V

Pin Name	Pin #	I/O
CAS4#	143	O
CAS1#	144	O
CAS5#	145	O
CAS2#	146	O
V_{DD}	147	V
CAS6#	148	O
CAS3#	149	O
CAS7#	150	O
WE#	151	O
PREQ1# / HDEV#	152	I
PREQ0#	153	I
PGNT1# / HRDY#	154	I/O
V_{SS}	155	V
PGNT0#	156	O
AD31	157	I/O
AD30	158	I/O
AD29	159	I/O
AD28	160	I/O
AD27	161	I/O
AD26	162	I/O
AD25	163	I/O
V_{SS}	164	V

Pin Name	Pin #	I/O
V_{DD}	165	V
AD24	166	I/O
C/BE3#	167	I/O
AD23	168	I/O
AD22	169	I/O
AD21	170	I/O
AD20	171	I/O
V_{SS}	172	V
AD19	173	I/O
AD18	174	I/O
AD17	175	I/O
AD16	176	I/O
C/BE2#	177	I/O
V_{SS}	178	V
FRAME#	179	s/t/s
IRDY#	180	s/t/s
TRDY#	181	s/t/s
KBDRST#	182	I
DEVSEL#	183	s/t/s
STOP#	184	s/t/s
LOCK#	185	I
PAR	186	I/O

Pin Name	Pin #	I/O
V_{SS}	187	V
C/BE1#	188	I/O
AD15	189	I/O
AD14	190	I/O
AD13	191	I/O
AD12	192	I/O
V_{SS}	193	V
AD11	194	I/O
AD10	195	I/O
AD9	196	I/O
AD8	197	I/O
C/BE0#	198	I/O
AD7	199	I/O
V_{SS}	200	V
V_{DD}	201	V
AD6	202	I/O
AD5	203	I/O
AD4	204	I/O
AD3	205	I/O
AD2	206	I/O
AD1	207	I/O
AD0	208	I/O

1.2 IB Pin Assignment

The IB package is a 160-pin Quad Flatpack (QFP). Figure 3 shows the package pin assignment. Table 3 and Table 4 list the pin assignment alphabetically and numerically, respectively.

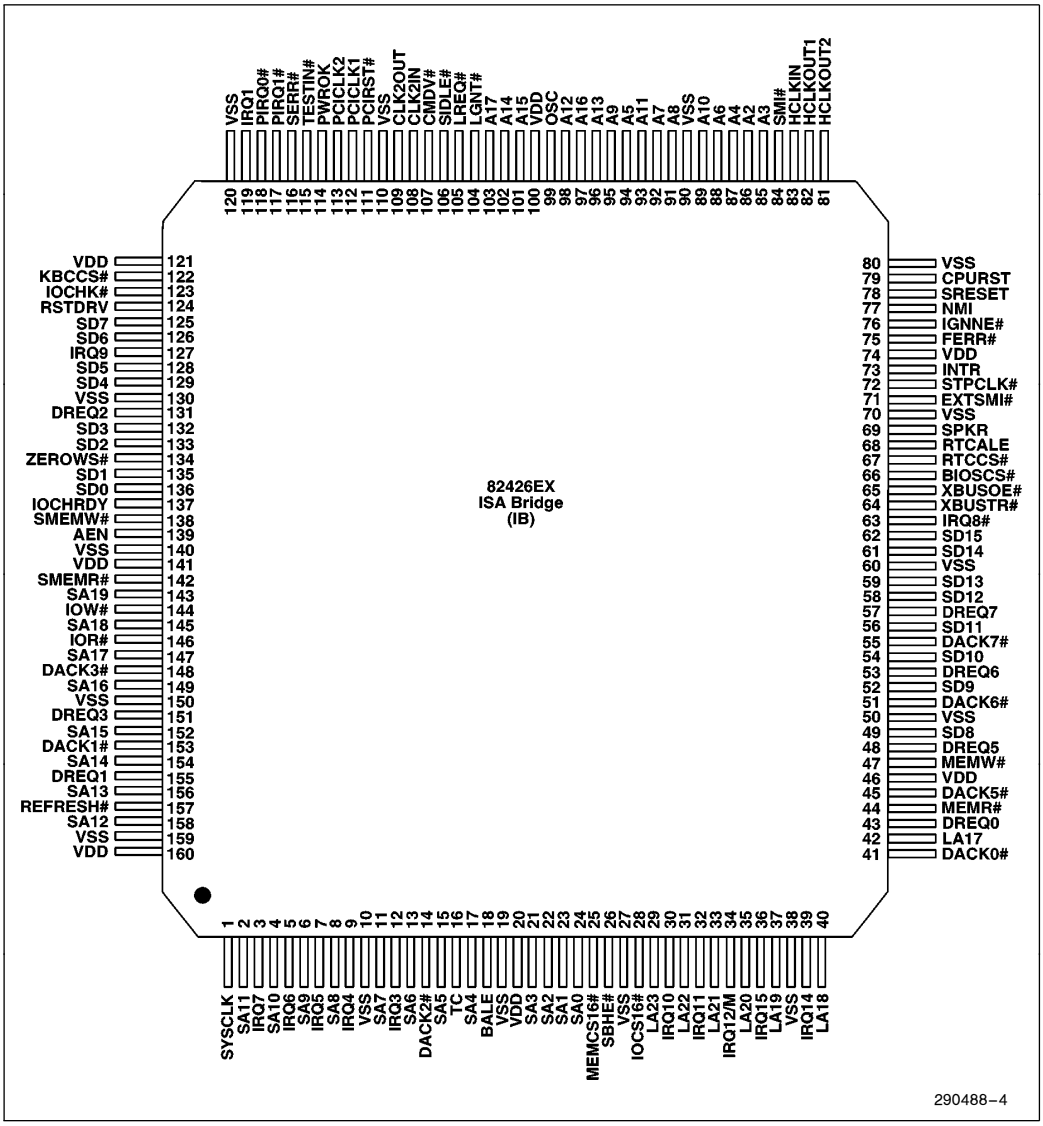


Figure 3. IB Pin Assignment

Table 3. Alphabetical IB Pin Assignment List

Pin Name	Pin #	I/O
A2	86	I/O
A3	85	I/O
A4	87	I/O
A5	94	I/O
A6	88	I/O
A7	92	I/O
A8	91	I/O
A9	95	I/O
A10	89	I/O
A11	93	I/O
A12	98	I/O
A13	96	I/O
A14	102	I/O
A15	101	I/O
A16	97	I/O
A17	103	I/O
AEN	139	O
BALE	18	O
BIOSCS#	66	O
CLK2IN	108	I
CLK2OUT	109	O
CMDV#	107	I/O
CPURST	79	O
DACK0#	41	O
DACK1#	153	O
DACK2#	14	O
DACK3#	148	O
DACK5#	45	O
DACK6#	51	O
DACK7#	55	O

Pin Name	Pin #	I/O
DREQ0	43	I
DREQ1	155	I
DREQ2	131	I
DREQ3	151	I
DREQ5	48	I
DREQ6	53	I
DREQ7	57	I
EXTSMI#	71	IS
FERR#	75	I
HCLKIN	83	I
HCLKOUT1	82	O
HCLKOUT2	81	O
IGNNE#	76	O
INTR	73	O
IOCHK#	123	I
IOCHRDY	137	I/O
IOCS16#	28	I
IOR#	146	I/O
IOW#	144	I/O
IRQ1	119	I
IRQ3	12	I
IRQ4	9	I
IRQ5	7	I
IRQ6	5	I
IRQ7	3	I
IRQ8#	63	I
IRQ9	127	I
IRQ10	30	I
IRQ11	32	I
IRQ12/M	34	I

Pin Name	Pin #	I/O
IRQ14	39	I
IRQ15	36	I
KBCCS#	122	O
LA17	42	I/O
LA18	40	I/O
LA19	37	I/O
LA20	35	I/O
LA21	33	I/O
LA22	31	I/O
LA23	29	I/O
LGNT#	104	I
LREQ#	105	O
MEMCS16#	25	I/O
MEMR#	44	I/O
MEMW#	47	I/O
NMI	77	O
OSC	99	I
PCICLK1	112	O
PCICLK2	113	O
PCIRST#	111	O
PIRQ0#	118	I
PIRQ1#	117	I
PWROK	114	IS
REFRESH#	157	I/O
RSTDRV	124	O
RTCALE	68	O
RTCCS#	67	O
SA0	24	I/O
SA1	23	I/O
SA2	22	I/O

Pin Name	Pin #	I/O
SA3	21	I/O
SA4	17	I/O
SA5	15	I/O
SA6	13	I/O
SA7	11	I/O
SA8	8	I/O
SA9	6	I/O
SA10	4	I/O
SA11	2	I/O
SA12	158	I/O
SA13	156	I/O
SA14	154	I/O
SA15	152	I/O
SA16	149	I/O
SA17	147	I/O
SA18	145	I/O
SA19	143	I/O
SBHE#	26	I/O
SD0	136	I/O
SD1	135	I/O
SD2	133	I/O
SD3	132	I/O
SD4	129	I/O
SD5	128	I/O
SD6	126	I/O
SD7	125	I/O
SD8	49	I/O
SD9	52	I/O
SD10	54	I/O
SD11	56	I/O

Table 3. Alphabetical IB Pin Assignment List (Continued)

Pin Name	Pin #	I/O
SD12	58	I/O
SD13	59	I/O
SD14	61	I/O
SD15	62	I/O
SERR#	116	I
SIDLE#	106	I/O
SMEMR#	142	O
SMEMW#	138	O
SMI#	84	O
SPKR	69	O

Pin Name	Pin #	I/O
SRESET	78	I
STPCLK#	72	O
SYSCLK	1	O
TC	16	O
TESTIN#	115	I
V_{DD}	20	V
V_{DD}	46	V
V_{DD}	74	V
V_{DD}	100	V
V_{DD}	121	V

Pin Name	Pin #	I/O
V_{DD}	141	V
V_{DD}	160	V
V_{SS}	10	V
V_{SS}	19	V
V_{SS}	27	V
V_{SS}	38	V
V_{SS}	50	V
V_{SS}	60	V
V_{SS}	70	V
V_{SS}	80	V

Pin Name	Pin #	I/O
V_{SS}	90	V
V_{SS}	110	V
V_{SS}	120	V
V_{SS}	130	V
V_{SS}	140	V
V_{SS}	150	V
V_{SS}	159	V
XBUSOE#	65	O
XBUSTR#	64	O
ZEROWS#	134	I

Table 4. Numerical IB Pin Assignment List

Pin Name	Pin #	I/O	Pin Name	Pin #	I/O	Pin Name	Pin #	I/O	Pin Name	Pin #	I/O
SYSCLK	1	O	LA22	31	I/O	SD14	61	I/O	A8	91	I/O
SA11	2	I/O	IRQ11	32	I	SD15	62	I/O	A7	92	I/O
IRQ7	3	I	LA21	33	I/O	IRQ8#	63	I	A11	93	I/O
SA10	4	I/O	IRQ12/M	34	I	XBUSTR#	64	O	A5	94	I/O
IRQ6	5	I	LA20	35	I/O	XBUSOE#	65	O	A9	95	I/O
SA9	6	I/O	IRQ15	36	I	BIOSCS#	66	O	A13	96	I/O
IRQ5	7	I	LA19	37	I/O	RTCCS#	67	O	A16	97	I/O
SA8	8	I/O	V_{SS}	38	V	RTCALE	68	O	A12	98	I/O
IRQ4	9	I	IRQ14	39	I	SPKR	69	O	OSC	99	I
V_{SS}	10	V	LA18	40	I/O	V_{SS}	70	V	V_{DD}	100	V
SA7	11	I/O	DACK0#	41	O	EXTSMI#	71	I	A15	101	I/O
IRQ3	12	I	LA17	42	I/O	STPCLK#	72	O	A14	102	I/O
SA6	13	I/O	DREQ0	43	I	INTR	73	O	A17	103	I/O
DACK2#	14	O	MEMR#	44	I/O	V_{DD}	74	V	LGNT#	104	I
SA5	15	I/O	DACK5#	45	O	FERR#	75	I	LREQ#	105	O
TC	16	O	V_{DD}	46	V	IGNNE#	76	O	SIDLE#	106	I/O
SA4	17	I/O	MEMW#	47	I/O	NMI	77	O	CMDV#	107	I/O
BALE	18	O	DREQ5	48	I	SRESET	78	I	CLK2IN	108	I
V_{SS}	19	V	SD8	49	I/O	CPURST	79	O	CLK2OUT	109	O
V_{DD}	20	V	V_{SS}	50	V	V_{SS}	80	V	V_{SS}	110	V
SA3	21	I/O	DACK6#	51	O	HCLKOUT2	81	O	PCIRST#	111	O
SA2	22	I/O	SD9	52	I/O	HCLKOUT1	82	O	PCICLK1	112	O
SA1	23	I/O	DREQ6	53	I	HCLKIN	83	I	PCICLK2	113	O
SA0	24	I/O	SD10	54	I/O	SMI#	84	O	PWROK	114	I
MEMCS16#	25	I/O	DACK7#	55	O	A3	85	I/O	TESTIN#	115	I
SBHE#	26	I/O	SD11	56	I/O	A2	86	I/O	SERR#	116	I
V_{SS}	27	V	DREQ7	57	I	A4	87	I/O	PIRQ1#	117	I
IOCS16#	28	I	SD12	58	I/O	A6	88	I/O	PIRQ0#	118	I
LA23	29	I/O	SD13	59	I/O	A10	89	I/O	IRQ1	119	I
IRQ10	30	I	V_{SS}	60	V	V_{SS}	90	V	V_{SS}	120	V

Table 4. Numerical IB Pin Assignment List (Continued)

Pin Name	Pin #	I/O
V_{DD}	121	V
KBCCS#	122	O
IOCHK#	123	I
RSTDRV	124	O
SD7	125	I/O
SD6	126	I/O
IRQ9	127	I
SD5	128	I/O
SD4	129	I/O
V_{SS}	130	V

Pin Name	Pin #	I/O
DREQ2	131	I
SD3	132	I/O
SD2	133	I/O
ZEROWS#	134	I
SD1	135	I/O
SD0	136	I/O
IOCHRDY	137	I/O
SMEMW#	138	O
AEN	139	O
V_{SS}	140	V

Pin Name	Pin #	I/O
V_{DD}	141	V
SMEMR#	142	O
SA19	143	I/O
IOW#	144	I/O
SA18	145	I/O
IOR#	146	I/O
SA17	147	I/O
DACK3#	148	O
SA16	149	I/O
V_{SS}	150	V

Pin Name	Pin #	I/O
DREQ3	151	I
SA15	152	I/O
DACK1#	153	O
SA14	154	I/O
DREQ1	155	I
SA13	156	I/O
REFRESH#	157	I/O
SA12	158	I/O
V_{SS}	159	V
V_{DD}	160	V

2.0 SIGNAL DESCRIPTION

This section contains a detailed description of each signal. The PSC signals are presented first, followed by the IB signals. The signals are arranged in functional groups according to their interface.

Note that the "#" symbol at the end of a signal name indicates that the active, or asserted state occurs when the signal is at a low voltage level. When "#" is not present after the signal name, the signal is asserted when at the high voltage level.

The terms assertion and negation are used extensively. This is done to avoid confusion when working with a mixture of "active-low" and "active-high" signals. The term **assert**, or **assertion** indicates that a signal is active, independent of whether that level is represented by a high or low voltage. The term **negate**, or **negation** indicates that a signal is inactive.

The following notations are used to describe signal types.

Signal Type	Description
I	**Input.** Standard input-only signal.
IS	**Input.** Schmitt Trigger
O	**Totem Pole Output.** Standard active driver.
OD	**Open Drain.** Input/Output
I/O	**Input/Output.** Bi-directional, tri-state pin.
s/t/s	**Sustained Tri-state.** Active low, tri-state signal with a pullup. Must be driven high for a clock before tri-state. Turn-around time must be maintained.

2.1 PSC Signals

2.1.1 HOST CPU INTERFACE SIGNALS (PSC)

Name	Type	Description
SRESET/ INIT	O	**SOFT RESET/INITIALIZE:** This is the soft reset output of the PSC and should be connected to the SRESET or INIT input to the CPU, depending on the CPU type.
A[31:30, 26:2]	I/O	**HOST ADDRESS:** A[31:30,26:2] are used as inputs to the PSC for CPU driven cycles. A[31:30,26:4] are outputs during Snoop cycles. Note that A[29:27] are not driven by the PSC. These signal lines must be externally driven low by either weak pull-down resistors or by driving these lines low when HLDA is active asserted. A[17:2] are also used for PSC/IB Link Interface transfers. These signals are tri-stated after a hard reset.
HD[31:0]	I/O	**HOST DATA:** HD[31:0] are connected to the host CPU data bus. These signals are inputs after a hard reset.
HDP[3:0]	I/O	**HOST DATA PARITY:** HP[3:0] are bi-directional parity signals for the host data bus. These signals provide parity to the PSC during main memory read cycles. The PSC sends parity information to main memory during non-CPU main memory write cycles. These signals are tri-stated after a hard reset.
BE[3:0]#	I/O	**BYTE ENABLE:** The Byte Enable signals indicate active bytes during read and write cycles. These signals are tri-stated after a hard reset.
M/IO# D/C# W/R#	I/O I I/O	**BUS CYCLE DEFINITION (Memory/Input-Output, Data/Code, Write/Read:** These signals define the Host Bus cycle. Note that special cycles are identified by BE[3:0]# and A[4:2]. These signals are tri-stated after a hard reset.

2.1.1 HOST CPU INTERFACE SIGNALS (PSC) (Continued)

Name	Type	Description
PCD/ CACHE#	I	**PAGE CACHE DISABLE/CACHE:** This multiplexed signal pin has two functions, depending on the type of CPU used. The **PCD** input signal, when asserted, indicates the current cycle can not be cached in the L2 cache during a cache line fill operation. When PCD is asserted the line will not be cached in L1 or L2. The **CACHE#** signal is active along with the first ADS# until the first RDY# or BRDY#. For line fills, the functionality of the CACHE# signal is identical to that of the PCD signal. During write-back cycles, CACHE# is always asserted at the beginning of the line write-back. The beginning of a write-back cycle is uniquely identified by active ADS#, W/R# and CACHE#. Beginning of the snoop write-back is identified by the ADS#, W/R#, CACHE# and HITM# being active.
ADS#	I	**ADDRESS STATUS:** The ADS# input indicates that the bus cycle definition signals (M/IO#, D/C#, W/R#), BE[3:0]#, and A[31:30, 26:2] are available on their corresponding pins.
RDY#	O	**READY:** RDY# indicates that the current non-burst bus cycle is complete. This signal is negated after a hard reset.
BRDY#	O	**BURST READY:** BRDY performs the same function during a burst cycle that RDY# performs during a non-burst cycle. This signal is negated after hard reset.
BLAST#	I	**BURST LAST:** BLAST# indicates the end of a burst access for CPU-initiated cycles.
HOLD	O	**HOLD:** The PSC asserts HOLD to the CPU to request ownership of the Host Bus. This signal is negated after a hard reset.
HLDA	I	**HOLDA:** HLDA must be asserted by the CPU for the PSC to grant a new master on the PCI or ISA Buses. When HLDA is negated, the CPU is the Host Bus master and the PSC is the PCI Bus master. When HLDA is negated, the PSC is also the master on the PSC/IB link interface.
AHOLD	O	**ADDRESS HOLD:** The AHOLD output signal forces the CPU to float its address bus in the next clock. The PSC asserts this signal in preparation to perform a PSC/IB Interface transfer, when SRESET needs to be asserted, or upon Deturbo logic requests. This signal is negated after a hard reset.
EADS#	O	**EXTERNAL ADDRESS:** EADS#, when asserted, indicates that an external address has been driven onto the CPU address lines. This address is used to perform an internal cache snoop cycle. This signal is negated after a hard reset.
KEN#	O	**CACHE ENABLE:** KEN#, when asserted, indicates whether the current cycle is cacheable in the CPU internal (L1 or primary) cache. This signal is negated after a hard reset.
HITM#	I	**HIT MODIFIED:** HITM#, when asserted, indicates that a hit to a modified data cache has occurred during the snoop cycle. A pull-up is used to keep HITM# negated, when not used.

2.1.2 SECONDARY CACHE SIGNALS (PSC)

Name	Type	Description
CI3E CI3O2	O	**CACHE INDEX SIGNALS:** The Cache Index signals generate the burst sequence required by the CPU during secondary cache accesses. The PSC latches the starting burst address and internally generates subsequent dword addresses for the entire cache line. The CI3E signal is always used for cache index bit 3. When used in a bank interleaved configuration, CI3O2 is used to drive cache index bit 3 to the odd bank, and CI3E is used to drive bit 3 to the even bank. When used in non-interleaved mode (only one bank), CI3O2 is used to drive cache index bit 2.
CWE[1:0]#	O	**CACHE WRITE ENABLE:** CWE[1:0]# are used to enable single writes and line fills to be written into the L2 cache. CWE0# is driven to all SRAMs in the even bank and CWE1# is driven to all SRAMs in the odd bank. The chip select signals of the SRAMs are asserted based on the byte enable signals and the W/R# signal, which are gated externally. Thus, only the bytes selected by the CPU are written. When programmed for non-interleaved mode, CWE[1:0]# mirror each other and are asserted to both banks. These signals are negated after a hard reset.
COE[1:0]#	O	**CACHE OUTPUT ENABLE:** COE[1:0]# are used to perform read cycles from the cache data SRAMs. COE0# is connected to the output enable pins of the cache data SRAMs of the even bank. COE1# is connected to the output enable pins of the cache data SRAMs of the odd bank. When programmed for non-interleaved mode, COE[1:0]# mirror each other and are asserted to both banks. These signals are negated after a hard reset.
TWE#	O	**TAG WRITE ENABLE:** TWE# is connected to the tag SRAM write enable (WE#) pin. TWE# is asserted during CPU read-miss cycles when a cache line is allocated and during write-hit cycles, when the Dirty (Modified) bit of the tag is updated. This signal is negated after a hard reset.
TAG[8:0]	I/O	**CACHE TAG:** TAG[8:0] are directly connected to the tag SRAM data bus. The L2 cache size determines the relationship between TAG[7:0] and the A[26:16] host address signals (see Section 4.0, Functional Description). TAG8 is always used as the Dirty (Modified) bit for the write-back L2 cache.

2.1.3 PCI SIGNALS (PSC)

Name	Type	Description
AD[31:0]	I/O	**ADDRESS/DATA:** AD[31:0] are connected to the PCI multiplexed address/data bus. These signals are also multiplexed with the IDE interface (refer to the section on PCI Bus IDE Signals). These signals are driven high after a hard reset.
C/BE[3:0]#	I/O	**BUS COMMAND/BYTE ENABLE:** PCI Bus commands (C) and Byte Enables (BE[3:0]#) are multiplexed on the same pins. PCI local bus command encoding and types are listed in Section 4.0, Functional Description. These signals are driven high after a hard reset.
FRAME#	I/O s/t/s	**FRAME:** FRAME# is an output when the PSC is a master on the PCI bus. FRAME indicates that a PCI cycle has started. This signal is tri-stated after a hard reset.
TRDY#	I/O s/t/s	**TARGET READY:** TRDY# is an input when PSC is a master on the PCI Bus. TRDY# is an output when the PSC acts as a PCI slave. TRDY# indicates that the target device is ready. This signal is tri-stated after a hard reset.
IRDY#	I/O s/t/s	**INITIATOR READY:** IRDY# is an output when PSC is a PCI master. IRDY# is an input when the PSC is a PCI slave. IRDY# indicates that the initiator of the cycle is ready. This signal is tri-stated after a hard reset.
LOCK#	I	**LOCK:** LOCK# indicates an exclusive bus operation and may require multiple transactions to complete. The PSC supports a bus type of LOCK only. Thus, when a PCI master locks the PCI Bus, it owns the system for the duration of the locked transactions.
STOP#	I/O s/t/s	**STOP:** STOP# indicates that the current bus target is requesting the master to stop the current transaction. STOP# is used for disconnect, retry, and abort sequences on the PCI Bus. This signal is tri-stated after a hard reset.
PAR	I/O	**PARITY:** PAR is driven by the PSC, as a PCI master, during the address and data phases for a write cycle and during the address phase for a read cycle. When the PSC is a PCI slave, parity is driven by the PSC for the data phase of a PCI read cycle. Parity is even parity across AD[31:0] and C/BE[3:0]#. PAR lags the corresponding address or data phase by 1 PCICLK. This signal is asserted after a hard reset.
SERR#	OC	**SYSTEM ERROR:** SERR#, when driven by the PSC, indicates that either a main memory parity error occurred or the PSC, as a master, received a target abort.
DEVSEL#	I/O s/t/s	**DEVICE SELECT:** DEVSEL#, when asserted, indicates that a PCI slave device has decoded the bus cycle address as the target of the current access. The PSC drives DEVSEL# based on the main memory address range being accessed by a PCI master. As an input, DEVSEL# indicates whether any device on the bus has been selected. This signal is tri-stated after a hard reset.

2.1.4 SYSTEM POWER MANAGEMENT (SMM) SIGNALS (PSC)

Name	Type	Description
SMI#	I	**SYSTEM MANAGEMENT INTERRUPT:** SMI#, when asserted, indicates that there is an active SMI#. It is used, along with SMIACT#, to block SRESET.
SMIACT#	I	**SYSTEM MANAGEMENT INTERRUPT ACTIVE:** SMIACT# indicates that the system is running in system management mode. SMIACT# is used by the PSC to access the SMRAM for CPU-initiated cycles. While SMIACT# is active, SRESET and A20M# must be blocked.

2.1.5 DRAM CONTROL SIGNALS (PSC)

Name	Type	Description
MA[10:0]	O	**MULTIPLEXED DRAM ADDRESS:** The MA[10:0] bus provides row and column address information to the main memory DRAMs.
RAS[4:0]#	O	**ROW ADDRESS STROBE:** Each of the RAS[4:0]# output signals corresponds to one DRAM row of four or eight bytes. These signals are used to latch the row addresses on the MA[10:0] bus into the DRAMs. RAS[4:0]# drive the DRAMs directly, without any external buffers. These signals are negated after a hard reset.
CAS[7:0]#	O	**COLUMN ADDRESS STROBE:** CAS[7:0]# are used to latch the column addresses on the MA[10:0] bus into the DRAMs. CAS[7:0]# drive the DRAMs directly, without any external buffers. These signals are negated after a hard reset.
WE#	O	**DRAM WRITE ENABLE:** WE# is externally buffered when MA[10:0] are externally buffered. This signal is asserted after a hard reset.

2.1.6 PSC/IB LINK INTERFACE (PSC)

See Section 2.2.7, PSC/IB Link Interface Signals (IB).

2.1.7 PCI BUS ARBITRATION/HOST BUS SLAVE DEVICE (PSC)

Name	Type	Description
PREQ1#/ HDEV#	I	**REQUEST1/HOST DEVICE:** This multiplexed signal has two functions. **PREQ1#** is used by the PCI master to gain control of the PCI Bus. This signal can be externally cascaded to support multiple PCI masters. The **HDEV#** function is used when the PSC is programmed to support a Host Bus slave device.
PREQ0#	I	**REQUEST0:** PREQ0# is used by PCI master to gain control of the PCI Bus. This signal can be externally cascaded to support multiple PCI masters.
PGNT1# HRDY#	I/O	**GRANT1/HOST READY: PGNT1#** is driven by the PSC to grant control of the PCI Bus to a PCI master. PGNT1# can be externally cascaded to support multiple PCI masters. The **HRDY#** function is used when the PSC is programmed to support a Host Bus slave device. This signal is driven high during and after a hard reset.
PGNT0#	O	**GRANT0:** PGNT0# is driven by the PSC to grant control of the PCI Bus to a PCI master. PGNT0# can be externally cascaded to support multiple PCI masters. This signal is driven high during and after a hard reset.

2.1.8 PCI BUS IDE (PSC)

LBIDE# is the only signal dedicated to PCI Bus IDE support. This pin is used to control the output enable of the 245 data transceivers and 244 control signal buffer. The other signals that support the IDE are shared with the PCI AD lines.

PCI Signal ADName	PCI IDE Signal Name	Type	Description
NA	LBIDE#	O	**LOCAL BUS IDE:** LBIDE# controls the output enables of the data transceivers and control signal buffers during accesses to the PCI local bus IDE path.
AD31	IDE1CS#	O	**IDE 1XX CHIP SELECT:** When the primary PCI local bus IDE is enabled, this signal is asserted for accesses to I/O addresses 1F0–1F7h. When the secondary PCI local bus IDE is enabled, this signal is asserted for an I/O cycle to addresses 170–177h.
AD30	IDE3CS#	O	**IDE 3XX CHIP SELECT:** When the primary PCI local bus IDE is enabled, this signal is asserted for accesses to I/O addresses 3F6,3F7h. When the secondary PCI local bus IDE is enabled this signal is asserted for accesses to I/O addresses 376,377h.
AD29	DIR	O	**DIRECTION:** DIR controls the direction of the data transceivers connected to the IDE connector. This signal is driven high for IDE reads and low for IDE writes.
AD28	IORDY	I	**IO READY:** IORDY allows the IDE drive to extend the cycle. The IDE cycle is held in wait-states as long as IORDY is sampled low. This input is synchronous to the first sample point, but asynchronous to subsequent sample points.
AD(27:25)	IDEA(2:0)	O	**IDE ADDRESS [2:0]:** These outputs are the A[2:0] signals that select individual ports on the IDE drive.
AD24	IOR#	O	**I/O READ STROBE:** IOR# is asserted for PCI local bus IDE I/O read cycles.
AD23	IOW#	O	**I/O WRITE STROBE:** IOW# is asserted for PCI local bus IDE I/O write cycles.
AD(15:0)	IDED(15:0)	I/O	**IDE DATA:** These bi-directional signals output data during IDE write cycles and input data during PCI local bus IDE read cycles.

2.1.9 CLOCKS AND RESET (PSC)

Name	Type	Description
PCICLKIN	I	**PCI CLOCK INPUT:** PCICLKIN is the clock input used by the PCI interface. PCI clock frequency can be configured to be the same or half the Host Bus frequency.
HCLKIN	I	**HOST BUS CLOCK INPUT:** HCLKIN is used for the Host Bus, L2 cache, PSC/IB Link and DRAM interfaces. Host Bus frequency can be configured to be the same or twice the PCI clock frequency.
CLK2IN	I	**CLOCK 2 INPUT:** CLK2IN is a 2X clock that is used during L2 cache writes.
CPURST	I	**CPU RESET:** CPURST is used as an input to place the PSC in a known state. CPURST is driven by the IB when PWROK is negated or when hard reset is driven by software through the TRC Register.
KBDRST#	I	**KEYBOARD RESET:** This signal is an input from the keyboard controller and is used to generate a soft reset to the CPU.

2.2 IB Signals

2.2.1 ISA INTERFACE SIGNALS (IB)

Name	Type	Description
BALE	O	**BUS ADDRESS LATCH ENABLE:** BALE is asserted by the IB to indicate that the address (SA[19:0], LA[23:17]) and SBHE# signal lines are valid. This signal is negated after a hard reset.
AEN	O	**ADDRESS ENABLE:** AEN is asserted during DMA cycles to prevent I/O slaves from mis-interpreting DMA cycles as valid I/O cycles. This signal is also asserted during IB-initiated refresh cycles. This signal is negated after a hard reset.
SYSCLK	O	**SYSTEM CLOCK:** Refer to the Clock signal descriptions.
IOCHRDY	I/O	**I/O CHANNEL READY:** Resources on the ISA Bus negate IOCHRDY to indicate that additional time (wait-states) is required to complete the cycle. This signal is normally high on the ISA Bus. IOCHRDY is an input when the IB owns the ISA Bus and the CPU or a PCI agent is accessing an ISA slave or during DMA transfers. IOCHRDY is output when an external ISA Bus master owns the ISA Bus and is accessing main memory or an IB register. As an IB output, IOCHRDY is negated from the falling edge of the ISA commands. After data is available for an ISA master read or the IB latches the data for a write cycle, IOCHRDY is asserted for 70 ns. After 70 ns, the IB tri-states IOCHRDY. The IB does not drive this signal when an ISA Bus master is accessing an ISA Bus slave. IOCHRDY is tri-stated upon CPURST.
IOCS16#	I	**16-BIT I/O CHIP SELECT:** This signal is driven by I/O devices on the ISA Bus to indicate that they support 16-bit I/O bus cycles.
IOCHK#	I	**I/O CHANNEL CHECK:** IOCHK# can be driven by any resource on the ISA Bus. When asserted, it indicates that a parity or an uncorrectable error has occurred for a device or memory on the ISA Bus. If IOCHK# is asserted and NMIs are enabled (via the NMISC and NMIERTC Registers), an NMI is generated to the CPU.

2.2.1 ISA INTERFACE SIGNALS (IB) (Continued)

Name	Type	Description
IOR#	I/O	**I/O READ:** IOR# asserted indicates to an ISA I/O slave device that the slave may drive data on to the ISA data bus (SD[15:0]). The I/O slave device must hold the data valid until after IOR# is negated. IOR# is an output when the IB owns the ISA Bus. IOR# is an input when an external ISA master owns the ISA Bus. This signal is negated after a hard reset.
IOW#	I/O	**I/O WRITE:** IOW# asserted indicates to an ISA I/O slave device that the slave may latch data from the ISA data bus (SD[15:0]). IOW# is an output when the IB owns the ISA Bus. IOW# is an input when an external ISA master owns the ISA Bus. This signal is negated after a hard reset.
LA[23:17]	I/O	**UNLATCHED ADDRESS:** These bi-directional address lines allow accesses to physical memory on the ISA Bus up to 16 MBytes. LA[23:17] are outputs when the IB owns the ISA Bus. The LA[23:17] lines become inputs when an ISA master owns the ISA Bus. The LA[23:17] signals are driven to an unknown state after a hard reset.
SA[19:0]	I/O	**SYSTEM ADDRESS BUS:** SA[19:0] are outputs when the IB owns the ISA Bus. SA[19:0] are inputs when an external ISA master owns the ISA Bus. Note that SA[19:17] have the same values as LA[19:17] for all memory cycles. For I/O accesses, only SA[15:0] are used. SA[19:0] are driven to an unknown state after a hard reset.
SBHE#	I/O	**SYSTEM BYTE HIGH ENABLE:** SBHE# indicates, when asserted, that a byte is being transferred on the upper byte (SD[15:8]) of the data bus. SBHE# is negated during refresh cycles. SBHE# is an output when the IB owns the ISA Bus and an input when an external ISA master owns the ISA Bus. This signal is at an unknown state after a hard reset .
MEMCS16#	OD	**MEMORY CHIP SELECT 16:** ISA slaves that are 16-bit memory devices drive this signal low. MEMCS16# is an input when the IB owns the ISA Bus. MEMCS16# is an output when an ISA Bus master owns the ISA Bus. The IB drives this signal low during ISA master to main memory cycles.
MEMR#	I/O	**MEMORY READ:** MEMR# is the command to a memory slave that it may drive data onto the ISA data bus. MEMR# is an output when the IB is a master on the ISA Bus and an input when an ISA master, other than the IB, owns the ISA Bus. This signal is also driven by the IB during refresh cycles. For DMA cycles, the IB, as a master, asserts MEMR#. This signal is tri-stated after a hard reset.
MEMW#	I/O	**MEMORY WRITE:** MEMW# is the command to a memory slave that it may latch data from the ISA data bus. MEMW# is an output when the IB owns the ISA Bus and an input when an ISA master, other than the IB, owns the ISA Bus. For DMA cycles, the IB, as a master, asserts MEMW#. This signal is tri-stated after a hard reset.
SMEMR#	O	**STANDARD MEMORY READ:** The IB asserts SMEMR# to request an ISA memory slave to drive data onto the data lines. If the access is below the 1 MByte range (00000000–000FFFFFh) during DMA compatible, IB master, or ISA master cycles, the IB asserts SMEMR#. SMEMR# is a delayed version of MEMR#. This signal is negated after a hard reset.
SMEMW#	O	**STANDARD MEMORY WRITE:** The IB asserts SMEMW# to request an ISA memory slave to accept data from the data lines. If the access is below the 1 MByte range (00000000–000FFFFFh) during DMA compatible, IB master, or ISA master cycles, the IB asserts SMEMW#. SMEMW# is a delayed version of MEMW#. This signal is negated after a hard reset.

2.2.1 ISA INTERFACE SIGNALS (IB) (Continued)

Name	Type	Description
ZEROWS#	I	**ZERO WAIT-STATES:** An ISA slave asserts ZEROWS# after its address and command signals have been decoded to indicate that the current cycle can be shortened. If IOCHRDY is negated and ZEROWS# is asserted during the same clock, then ZEROWS# is ignored and wait-states are added as a function of IOCHRDY (i.e., IOCHRDY has precedence over ZEROWS#).
SD[15:0]	I/O	**SYSTEM DATA:** SD[15:0] provide the 16-bit data path for devices residing on the ISA Bus. SD[15:8] correspond to the high order byte and SD[7:0] correspond to the low order byte. SD[15:0] are undefined during refresh. These signals are tri-stated after a hard reset.

2.2.2 NMI SIGNALS (IB)

Name	Type	Description
NMI	O	**NON-MASKABLE INTERRUPT:** NMI is used to force a non-maskable interrupt to the CPU. The IB generates an NMI when either SERR# or IOCHK# is asserted, depending on how the NMI Status and Control Register is programmed. NMI generation can be globally disabled. This signal is negated after a hard reset.
SERR#	I	**SYSTEM ERROR:** SERR# can be pulsed active by any PCI device that detects a system error condition. Upon sampling SERR# active, the IB generates a non-maskable interrupt (NMI) to the CPU.

2.2.3 DMA SIGNALS (IB)

Name	Type	Description
DREQ [3:0,7:5]	I	**DMA REQUEST:** The DREQ lines are used to request DMA service from the IB's DMA controller or for a 16-bit master to gain control of the ISA expansion bus. The active level (high or low) is programmed via the DMA Command Register. All inactive to active edges of DREQ are assumed to be asynchronous. The request must remain active until the appropriate DACKx# signal is asserted.
DACK# [3:0, 7:5]	O	**DMA ACKNOWLEDGE:** The DACK output lines indicate that a request for DMA service has been granted by the IB or that a 16-bit master has been granted the bus. The active level (high or low) is programmed via the DMA Command Register. These signals are negated after a hard reset.
TC	O	**TERMINAL COUNT:** The IB asserts TC to DMA slaves as a terminal count indicator. This signal is negated after a hard reset.
REFRESH#	I/O	**REFRESH:** As an output, REFRESH# asserted indicates when a refresh cycle is in progress. As an output, this signal is driven directly onto the ISA Bus. This signal is an output only when the IB DMA refresh controller is a master on the bus responding to an internally generated request for refresh. As an input, REFRESH# is driven by 16-bit ISA Bus masters to initiate refresh cycles. This signal is tri-stated after a hard reset.

2.2.4 TIMER/COUNTER SIGNALS (IB)

Name	Type	Description
SPKR	O	**SPEAKER DRIVE:** This signal drives an external speaker driver device, which in turn drives the ISA system speaker. This signal can be enabled/disabled via the NMI Status and Control Register. When enabled, the SPKR signal is the output of counter 2. This signal is negated after a hard reset.
OSC	I	**OSCILLATOR:** The oscillator is the 14.31818 MHz ISA clock signal. It is used by the internal 82C54 Timer, counters 0, 1, and 2.

2.2.5 INTERRUPT CONTROLLER SIGNALS (IB)

Name	Type	Description
IRQ[15,14, 11:9,7:3,1]	IS	**INTERRUPT REQUEST:** The IRQ signals provide both system board components and ISA Bus I/O devices with a mechanism for asynchronously interrupting the CPU. The assertion mode (edge or level triggered) is selected via the Edge/Level Triggered 1 Register and Edge/Level Triggered 2 Register. For edge triggered mode, a low-to-high transition on the IRQ line is recognized as an interrupt request. For level triggered mode, a low level on the IRQ line is recognized as an interrupt request. IRQ[8#,2:0] and the internal IRQ13 (FERR#) are not programmable through the ELCR registers. These IRQ's, with the exception of IRQ8#, are always active high edge triggered and can not be modified by software. IRQ8# is always active low edge sensitive. A low-to-high transition on IRQ1 is latched by the IB. The IB continues to generate an internal IRQ1 to the 8259 core until a CPURST or an I/O read access to port 60h is detected. These signals are placed in edge triggered mode after a hard reset.
IRQ8#	IS	**INTERRUPT REQUEST 8:** IRQ8# is always an active low edge triggered interrupt input (i.e. this interrupt can not be modified by software).
IRQ12/M	IS	**INTERRUPT REQUEST 12/MOUSE INTERRUPT:** Refer to the X-Bus Signal Description.
PIRQ[0,1]#	IS	**PROGRAMMABLE INTERRUPT REQUEST:** The PIRQ0 and PIRQ1 signals can be shared with interrupts IRQ[15,14,12:9,7:3]. The routing is controlled by the PIRQ Route Control Registers. Each PIRQx# line has a separate Route Control Register. These signals require external pull-up resistors.
INTR	O	**CPU INTERRUPT:** The IB asserts INTR to signal the CPU that an interrupt request is pending and needs to be serviced. It is asynchronous with respect to SYSCLK or HCLKIN. The interrupt controller must be programmed following reset to ensure that INTR is at a known state. This signal is negated after a hard reset.

2.2.6 X-BUS SIGNALS (IB)

Name	Type	Description
XBUSTR#	O	**X-BUS DATA TRANSMIT/RECEIVE:** XBUSTR# is tied directly to the direction control of a 74F245 that buffers the X-Bus data (XD[7:0]). XBUSTR# is asserted for all I/O read cycles, regardless if the access is to an IB supported device. XBUSTR# is asserted for memory cycles only if BIOS space has been decoded. This signal is negated after a hard reset.
XBUSOE#	O	**X-BUS OUTPUT ENABLE:** XBUSOE# is tied directly to the output enable of a 74F245 that buffers the X-Bus data (XD[7:0]), from the system data bus, SD[7:0]. XBUSOE# is asserted anytime an IB supported X-Bus device is decoded, and the devices decode is enabled in the X-Bus Chip Select Enable Register (XBCSA Register). This signal is negated after a hard reset.
KBCCS#	O	**KEYBOARD CONTROLLER CHIP SELECT:** This signal is asserted during read or write accesses to KBC locations 60h, 62h, 64h, or 66h. This signal is negated after a hard reset.
BIOSCS#	O	**BIOS CHIP SELECT:** This signal is asserted during read or write accesses to BIOS. During DMA cycles, BIOSCS# is not generated. This signal is negated after a hard reset.
RTCCS#	O	**REAL TIME CLOCK CHIP SELECT:** This signal is asserted during read or write accesses to RTC location 71h. RTCALE can be tied to a pair of external OR gates to generate the real time clock read and write command signals. This signal is negated after a hard reset.
RTCALE	O	**REAL TIME CLOCK ADDRESS LATCH:** RTCALE latches the appropriate memory address into the RTC. A write to port 70h with the appropriate RTC memory address that will be written to or read from, causes RTCALE to be asserted. RTCALE is asserted based on IOW# falling, and remains asserted for two SYSCLKs. This signal is negated after a hard reset.
IGNNE#	O	**IGNORE ERROR:** This signal is connected to the ignore error pin on the CPU. IGNNE# is only used if the IB coprocessor error reporting function is enabled in the XBCSA Register (bit 5 = 1). This signal is negated after a hard reset.
FERR#	IS	**NUMERIC COPROCESSOR ERROR:** This pin functions as a FERR# signal supporting coprocessor errors. This signal is tied to the coprocessor error signal on the CPU. FERR# is only used if the IB coprocessor error reporting function is enabled in the XBCSA Register (bit 5 = 1). FERR# has a weak internal pull-up resistor to ensure a high level when the coprocessor error function is disabled.
IRQ12/M	IS	**INTERRUPT REQUEST/MOUSE INTERRUPT:** In addition to providing the standard interrupt function as described in the signal description for IRQ[15,14, 11:9, 7:3, 1], the IRQ12/M signal also provides a mouse interrupt function. The X-Bus Chip Select Register determines the functionality of IRQ12/M. An internal IRQ12 interrupt continues to be generated until a reset or an I/O read access to address 60h (falling edge of IOR#) is detected. After a reset, this pin provides the standard IRQ12 function.

2.2.7 PSC/IB LINK INTERFACE SIGNALS (IB)

Name	Type	Description
CMDV#	I/O	**COMMAND VALID:** The link master (PSC or IB) asserts CMDV# to indicate the beginning of a link transfer. The PSC negates this signal after a hard reset. CMDV# is used along with SIDLE# to set the PSC/IB system clock configuration during a PWROK hard reset. These inputs are strapped to the appropriate levels, sampled while PWROK is inactive, and latched when PWROK goes active (see Section 4.0, Functional Description).
SIDLE#	I/O	**SLAVE IDLE:** The link slave (PSC or IB) asserts SIDLE# to indicate that it is available for data transfers. The IB asserts this signal after a hard reset. SIDLE# is used along with CMDV# to set the PSC/IB system clock configuration during PWROK hard reset. These inputs are strapped to the appropriate levels, sampled while PWROK is inactive, and latched when PWROK goes active (see Section 4.0, Functional Description).
LREQ#	O	**LINK REQUEST:** The IB asserts LREQ# to request a link transfer. This signal is negated after a hard reset.
LGNT#	I	**LINK GRANT:** The PSC asserts LGNT# to grant the IB a link transfer. This signal is negated after a hard reset.
A[17:2]	I/O	**HOST ADDRESS/LINK:** For PSC/IB Link transfers, A[17:2] transfer data/commands between the IB and PSC. These signals are tri-stated after a hard reset.

2.2.8 SYSTEM POWER MANAGEMENT (SMM) SIGNALS (IB)

Name	Type	Description
SMI#	O	**SYSTEM MANAGEMENT INTERRUPT:** SMI# is an active low synchronous output that is asserted by the IB in response to one of many enabled hardware or software events. SMI# is ORed externally with SRESET and driven to the CPU. This signal is negated after a hard reset.
STPCLK#	O	**STOP CLOCK:** STPCLK# is an active low synchronous output that is asserted by the IB in response to one of many enableable hardware or software events. STPCLK# connects directly to the CPU. This signal is negated after a hard reset.
EXTSMI#	I	**EXTERNAL SYSTEM MANAGEMENT INTERRUPT:** EXTSMI# is a falling edge triggered input to the IB indicating that an external device is requesting the system to enter SMM mode. When enabled, a falling edge on EXTSMI# results in the assertion of the SMI# signal to the CPU. EXTSMI# is an asynchronous input to the IB. However, when the setup and hold times are met it is only required to be asserted for one HCLKIN. Once negated it must remain negated for at least four HCLKINs in order to allow the edge detect logic to reset.

2.2.9 SYSTEM CLOCK SIGNALS (IB)

Name	Type	Description
CLK2IN	I	**2X CLOCK IN:** CLK2IN is a 2X clock input. CLK2IN is divided as shown in Section 4.0, Functional Description, to generate HCLKIN, PCICLK, and SYSCLK.
CLK2OUT	O	**2X Clock Out:** CLK2OUT is a delayed version of CLK2IN. The PSC uses this clock.
HCLKOUT[2,1]	O	**HOST CLOCK OUT:** HCLKOUT[2,1] provide the reference clock for the IB, PSC, and CPU devices. The IB divides the CLK2IN input to generate HCLKOUT[2,1]. Either HCLKOUT2 or HCLKOUT1 is routed back to the IB's HCLKIN input providing the IB with its HCLKIN reference.
HCLKIN	I	**HOST CLOCK IN:** This 1X clock input provides the fundamental timing and internal operating frequency for the IB. This signal is connected as a feedback from the HCLKOUT outputs. The IB operates at 25 MHz or 33 MHz, depending on the frequency of the CLK2IN input.
PCICLK[2,1]	O	**PCI CLOCK OUT:** PCICLK[2,1] provide the reference clock for the PSC and PCI devices. The IB divides the CLK2IN input to generate PCICLK[2,1]. The PCI Bus operates at 25 MHz or 33 MHz, depending on the frequency of the CLK2IN input.
SYSCLK	O	**ISA SYSTEM CLOCK:** SYSCLK is the reference clock for the ISA Bus. It drives the ISA Bus directly. The SYSCLK frequencies supported are 8 MHz and 8.33 MHz.

2.2.10 SYSTEM RESET SIGNALS (IB)

Name	Type	Description
PWROK	I	**POWER OK:** When asserted, PWROK is an indication to the IB that power and CLK2IN have been stable for at least 1 ms. PWROK can be driven asynchronously. When PWROK is negated, the IB asserts CPURST, PCIRST# and RSTDRV. The IB also uses the rising edge of PWROK to sample the CMDV# and SIDLE# signals to determine the HCLKIN, PCICLK, and SYSCLK divisor values.
CPURST	O	**CPU RESET:** The IB asserts CPURST to reset the CPU, IB, and PSC. The IB asserts CPURST during power-up and when a hard reset sequence is initiated through the TRC Register. If a hard reset is initiated through the TRC register, the IB resets it's internal registers and signals to their default state, but maintains its clock divisor values.
SRESET	I	**Soft Reset:** SRESET is used internally by the IB.
PCIRST#	O	**PCI RESET:** The IB asserts PCIRST# to reset devices that reside on the PCI bus. The IB asserts PCIRST# during power-up and when a hard reset sequence is initiated through the TRC register. PCIRST# is driven asynchronously relative to PCICLK.
RSTDRV	O	**RESET DRIVE:** The IB asserts RSTDRV to reset devices that reside on the ISA Bus. The IB asserts this signal during a hard reset and during power-up.

2.2.11 TEST SIGNALS (IB)

Name	Type	Description
TESTIN#	I	**TEST:** The TESTIN# signal is used to tri-state all of the IB outputs. During normal operation, this input should be pulled up.

3.0 REGISTER DESCRIPTION

The 82420EX PCIset contains I/O control registers, PCI configuration registers, and ISA Compatible registers. These registers are discussed in this section.

The PCIset, upon receiving a hard reset, sets its internal registers to pre-determined **default** states. The default values are indicated in the individual register descriptions. Note that the default state of some ISA-Compatible register bits is indeterminate after a hard reset.

The following notation is used to describe register access attributes:

RO **Read Only.** If a register is read only, writes have no effect.

WO **Write Only.** If a register is write only, reads have no effect.

R/W **Read/Write.** A register with this attribute can be read and written. Note that individual bits in some read/write registers may be read only.

3.1 Register Access

Table 5, Table 6, and Table 7 show the I/O assignments for the I/O Control Registers, PCI Configuration Registers, and the ISA-Compatible Registers. Little-endian ordering is used for all multi-byte accesses (i.e., lower addresses contain the least significant parts of the fields).

NOTE:
Aliasing of the 90–9Fh address range to 80–8Fh is enabled/disabled in the ISA Controller Recovery Timer Register. When aliasing is enabled, the IB aliases I/O accesses in the 90–9Fh range to the 80–8Fh range. In this case, the IB only forwards write accesses to these locations to the ISA Bus. When aliasing is disabled, the IB allows both read and write accesses to the 90–9Fh range to be forwarded to the ISA Bus (i.e. they are no longer considered IB internal registers). Note that port 92h is always a distinct ISA register in the 90–9Fh range and is always forwarded to the ISA Bus. In addition, when aliasing is disabled, ISA master accesses to the 90h–9Fh range are ignored by the IB.

I/O Control Registers

The I/O control registers (Table 5) are located in the CPU I/O space and can only be accessed by the CPU. The TRC and CONFDATA Registers can be accessed as byte, word, or dword quantities. The CONFADD Register can only be accessed as a dword quantity.

The CONFADD and CONFDATA Registers are used to access PCI configuration space. This is accomplished in two steps. First, the PCI configuration address is written to the CONFADD Register using the PCI configuration space access mechanism 1 address field definitions. Second, configuration register data is read/written from/to the CONFDATA Register address location.

The address written to the CONFADD Register contains five programmable fields (Bus Number, Device Number, Function Number, Configuration Register Offset, and the configuration enable bit—bit 31). If the Device Number = 05, the Bus Number = 00, and bit 31 = 1, subsequent CONFDATA Register accesses, read/write the PCI configuration register pointed to by the Register Offset field. If the Register Offset field points to a reserved register location, reads return all 0's to the CPU and writes are ignored by the PSC. If bit 31 = 1, but the Device Number\05, a PCI configuration cycle is run on the PCI Bus. If bit 31 = 0 (regardless of the Bus Number or Device Number values), the access to the CONFDATA Register location is forwarded to the PCI Bus and, if unclaimed on PCI, forwarded to ISA as a normal access to I/O address 0CFCh.

Table 5. I/O Control Registers

I/O Address	Mnemonic	Register	Register Access	Bus Master
0CF8h	CONFADD	Configuration Address	R/W	CPU Only
0CFCh	CONFDATA	Configuration Data	R/W	CPU Only
0CF9h	TRC	Turbo/Reset Control	R/W	CPU Only

Mechanism 1 PCI Configuration Address Fields

31	30 24	23 16	15 11	10 8	7 2	1 0
	Reserved	Bus Number	Device Number	Function Number	Register Offset	0 0

NOTE:
Device number = 05 is equivalent to IDSEL 16. Thus, other PCI devices cannot use IDSEL 16.

PCI Configuration Registers

The PCI configuration registers are located in the 82420EX PCIset's PCI configuration space and can only be accessed by the CPU. These registers (Table 6) can be accessed as byte, word, or dword quantities. The addresses for the configuration registers in the table are PCI configuration space offset values. The CPU accesses PCI configuration space (all PCI devices including the PSC) using mechanism 1 configuration access. For a detailed description of the PCI mechanism 1 configuration access, refer to the PCI Local Bus Specification document.

Some of the PCI configuration registers contain reserved bits. When reserved bits are read, a value of 0 is returned. In addition, the PCI configuration space includes reserved I/O locations. When reserved I/O locations are read, a value of 00h is returned. Writes to reserved bits or reserved I/O locations have no affect.

ISA-Compatible Registers

The ISA Compatible registers (Table 7) include DMA registers, timer registers, interrupt control registers, non-maskable interrupt, X-Bus support, and advanced power management control. These registers can be accessed by the CPU, a PCI master, or an ISA master as shown in Table 7. CPU or PCI masters can access the ISA-Compatible registers as 8-bit, 16-bit, 24-bit, or 32-bit quantities. However, only the first active BE[3:0]# is processed by the PSC. The remaining active byte enables in the same cycle are ignored. ISA Bus masters access the registers as 8-bit quantities. Unless otherwise stated in the individual register description, reserved bits must be written with a 0 and these bits return a 0 when read.

Table 6. PCI Configuration Register

Configuration Offset	Mnemonic	Register	Register Access	Bus Master
00–01h	VID	Vendor Identification	RO	CPU Only
02–03h	DID	Device Identification	RO	CPU Only
04–05h	PCICOM	PCI Command	R/W	CPU Only
06–07h	DS	Device Status	R/WC	CPU Only
08h	RID	Revision Identification	RO	CPU Only
09–3Fh	—	Reserved	—	—
40h	PCICON	PCI Control	R/W	CPU Only
41–43h	—	Reserved	—	—
44h	HDEVCON	Host Device Control	R/W	CPU Only
45–47h	—	Reserved	—	—
48–49h	LBIDE	PCI Local Bus IDE Control	R/W	CPU Only
4A–4Bh	—	Reserved	—	—
4Ch	IORT	ISA I/O Recovery Timer	R/W	CPU Only
4Dh	PREV	Part Revision Identification	R/W	CPU Only
4Eh	XBCSA	X-Bus Chip Select Enable A	R/W	CPU Only
4Fh	—	Reserved	—	—
50h	HOSTSEL	Host Bus Select	R/W	CPU Only
51h	DFC	Deturbo Frequency Control Register	R/W	CPU Only
52–53h	SCC	Secondary Cache Control	R/W	CPU Only
54–55h	—	Reserved	—	—
56–57h	DRAMC	DRAM Control	R/W	CPU Only
58h	—	Reserved	—	—

Table 6. PCI Configuration Register (Continued)

Configuration Offset	Mnemonic	Register	Register Access	Bus Master
59–5Fh	PAM	Programmable Attribute Map Registers	R/W	CPU Only
60–64h	DRB	DRAM Row Boundary Registers	R/W	CPU Only
65h	—	Reserved	—	—
66h	PIRQ0RC	PIRQ0 Route Control	R/W	CPU Only
67h	PIRQ1RC	PIRQ1 Route Control	R/W	CPU Only
68h	DMH	DRAM Memory Hole	R/W	CPU Only
69h	TOM	Top of Memory	R/W	CPU Only
6A–6Fh	—	Reserved	—	—
70h	SMRAMCON	SMRAM Control	R/W	CPU Only
71–9Fh	—	Reserved	—	—
A0h	SMICNTL	SMI Control	R/W	CPU Only
A1h	—	Reserved	R/W	CPU Only
A2–A3h	SMIEN	SMI Enable	R/W	CPU Only
A4–A7h	SEE	System Event Enable	R/W	CPU Only
A8h	FTMR	Fast Off Timer	R/W	CPU Only
A9	—	Reserved	—	—
AA–ABh	SMIREQ	SMI Request	R/W	CPU Only
ACh	CTLTMRL	Clock Throttle STPCLK# Low Timer	R/W	CPU Only
ADh	—	Reserved	—	—
AEh	CTLTMRH	Clock Throttle STPCLK# High Timer	R/W	CPU Only
AF–FFh	—	Reserved	—	—

Table 7. ISA-Compatible Registers

Address	FEDC	BA98	7654	3210	Register Name	Access Type	Bus Access
0000h	0000	0000	000x	0000	DMA1 CH0 Base and Current Address	r/w	CPU/PCI
0001h	0000	0000	000x	0001	DMA1 CH0 Base and Current Count	r/w	CPU/PCI
0002h	0000	0000	000x	0010	DMA1 CH1 Base and Current Address	r/w	CPU/PCI
0003h	0000	0000	000x	0011	DMA1 CH1 Base and Current Count	r/w	CPU/PCI
0004h	0000	0000	000x	0100	DMA1 CH2 Base and Current Address	r/w	CPU/PCI
0005h	0000	0000	000x	0101	DMA1 CH2 Base and Current Count	r/w	CPU/PCI
0006h	0000	0000	000x	0110	DMA1 CH3 Base and Current Address	r/w	CPU/PCI
0007h	0000	0000	000x	0111	DMA1 CH3 Base and Current Count	r/w	CPU/PCI
0008h	0000	0000	000x	1000	DMA1 Status (r), Command (w)	r/w	CPU/PCI
0009h	0000	0000	000x	1001	DMA1 Request	wo	CPU/PCI
000Ah	0000	0000	000x	1010	DMA1 Write Single Mask Bit	wo	CPU/PCI
000Bh	0000	0000	000x	1011	DMA1 Channel Mode	wo	CPU/PCI
000Ch	0000	0000	000x	1100	DMA1 Clear Byte Pointer	wo	CPU/PCI
000Dh	0000	0000	000x	1101	DMA1 Master Clear	wo	CPU/PCI
000Eh	0000	0000	000x	1110	DMA1 Clear Mask	wo	CPU/PCI
000Fh	0000	0000	000x	1111	DMA1 Write All Mask Bits	r/w	CPU/PCI
0020h	0000	0000	001x	xx00	INT 1 Control	r/w	CPU/PCI/ISA
0021h	0000	0000	001x	xx01	INT 1 Mask	r/w	CPU/PCI/ISA
0040h	0000	0000	010x	0000	Timer Counter 1 - Counter 0 Count	r/w	CPU/PCI/ISA
0041h	0000	0000	010x	0001	Timer Counter 1 - Counter 1 Count	r/w	CPU/PCI/ISA
0042h	0000	0000	010x	0010	Timer Counter 1 - Counter 2 Count	r/w	CPU/PCI/ISA
0043h	0000	0000	010x	0011	Timer Counter 1 Command Mode	wo	CPU/PCI/ISA
0060h[1]	0000	0000	0110	00x0	Reset X-Bus IRQ12/M and IRQ1	r	CPU/PCI/ISA
0061h	0000	0000	0110	0xx1	NMI Status and Control	r/w	CPU/PCI/ISA

Table 7. ISA-Compatible Registers (Continued)

Address	FEDC	BA98	7654	3210	Register Name	Access Type	Bus Access
0070h[1]	0000	0000	0111	0xx0	CMOS RAM Address and NMI Mask	wo	PCI/ISA
0080h[2]	0000	0000	100x	0000	DMA Page Register (Reserved)	r/w	CPU/PCI/ISA
0081h	0000	0000	100x	0001	DMA Channel 2 Page	r/w	CPU/PCI/ISA
0082h	0000	0000	1000	0010	DMA Channel 3 Page	r/w	CPU/PCI/ISA
0083h	0000	0000	100x	0011	DMA Channel 1 Page	r/w	CPU/PCI/ISA
0084h[2]	0000	0000	100x	0100	DMA Page Register (Reserved)	r/w	CPU/PCI/ISA
0085h[2]	0000	0000	100x	0101	DMA Page Register (Reserved)	r/w	CPU/PCI/ISA
0086h[2]	0000	0000	100x	0110	DMA Page Register (Reserved)	r/w	CPU/PCI/ISA
0087h	0000	0000	100x	0111	DMA Channel 0 Page	r/w	CPU/PCI/ISA
0088h[2]	0000	0000	100x	0100	DMA Page Register (Reserved)	r/w	CPU/PCI/ISA
0089h	0000	0000	100x	1001	DMA Channel 6 Page	r/w	CPU/PCI/ISA
008Ah	0000	0000	100x	1010	DMA Channel 7 Page	r/w	CPU/PCI/ISA
008Bh	0000	0000	100x	1011	DMA Channel 5 Page	r/w	CPU/PCI/ISA
008Ch[2]	0000	0000	100x	1100	DMA Page Register (Reserved)	r/w	CPU/PCI/ISA
008Dh[2]	0000	0000	100x	1101	DMA Page Register (Reserved)	r/w	CPU/PCI/ISA
008Eh[2]	0000	0000	100x	1110	DMA Page Register (Reserved)	r/w	CPU/PCI/ISA
008Fh	0000	0000	100x	1111	DMA Low Page Register Refresh	r/w	CPU/PCI/ISA
00A0h	0000	0000	101x	xx00	INT 2 Control	r/w	CPU/PCI/ISA
00A1h	0000	0000	101x	xx01	INT 2 Mask	r/w	CPU/PCI/ISA
00B2h	0000	0000	1011	0010	Advanced Power Management Control	r/w	CPU Only
00B3h	0000	0000	1011	0011	Advanced Power Management Status	r/w	CPU/PCI
00C0h	0000	0000	1100	000x	DMA2 CH0 Base and Current Address	r/w	CPU/PCI
00C2h	0000	0000	1100	001x	DMA2 CH0 Base and Current Count	r/w	CPU/PCI
00C4h	0000	0000	1100	010x	DMA2 CH1 Base and Current Address	r/w	CPU/PCI

Table 7. ISA-Compatible Registers (Continued)

Address	FEDC	BA98	7654	3210	Register Name	Access Type	Bus Access
00C6h	0000	0000	1100	011x	DMA2 CH1 Base and Current Count	r/w	CPU/PCI
00C8h	0000	0000	1100	100x	DMA2 CH2 Base and Current Address	r/w	CPU/PCI
00CAh	0000	0000	1100	101x	DMA2 CH2 Base and Current Count	r/w	CPU/PCI
00CCh	0000	0000	1100	110x	DMA2 CH3 Base and Current Address	r/w	CPU/PCI
00CEh	0000	0000	1100	111x	DMA2 CH3 Base and Current Count	r/w	CPU/PCI
00D0h	0000	0000	1101	000x	DMA2 Status(r) Command(w)	r/w	CPU/PCI
00D2h	0000	0000	1101	001x	DMA2 Request	wo	CPU/PCI
00D4h	0000	0000	1101	010x	DMA2 Write Single Mask Bit	wo	CPU/PCI
00D6h	0000	0000	1101	011x	DMA2 Write Mode Register	wo	CPU/PCI
00D8h	0000	0000	1101	100x	DMA2 Clear Byte Pointer	wo	CPU/PCI
00DAh	0000	0000	1101	101x	DMA2 Master Clear	wo	CPU/PCI
00DCh	0000	0000	1101	110x	DMA2 Clear Mask	wo	CPU/PCI
00DEh	0000	0000	1101	111x	DMA2 Write All Mask Bits	r/w	CPU/PCI
00F0h[1]	0000	0000	1111	0000	Coprocessor Error	wo	CPU/PCI/ISA
0481h	0000	0100	1000	0001	DMA CH2 High Page Register	r/w	CPU/PCI/ISA
0482h	0000	0100	1000	0010	DMA CH3 High Page Register	r/w	CPU/PCI/ISA
0483h	0000	0100	1000	0011	DMA CH1 High Page Register	r/w	CPU/PCI/ISA
0487h	0000	0100	1000	0111	DMA CH0 High Page Register	r/w	CPU/PCI/ISA
0489h	0000	0100	1000	1001	DMA CH6 High Page Register	r/w	CPU/PCI/ISA
048Ah	0000	0100	1000	1010	DMA CH7 High Page Register	r/w	CPU/PCI/ISA
048Bh	0000	0100	1000	1011	DMA CH5 High Page Register	r/w	CPU/PCI/ISA
04D0h	0000	0100	1101	0000	INT-1 Edge/Level Control	r/w	CPU/PCI/ISA
04D1h	0000	0100	1101	0001	INT-2 Edge/Level Control	r/w	CPU/PCI/ISA

NOTES:
1. Read and write accesses to these locations are always forwarded to the ISA Bus.
2. Write accesses to these locations are forwarded to the ISA Bus. Read Accesses are not forwarded to the ISA Bus. If programmed in the ISA I/O Recovery Timer Register, the IB will not alias the 90–9Fh address range with the 80–8Fh address range. In this case, accesses to the 90–9Fh address range are forwarded to the ISA Bus for both reads and writes (i.e. they are no longer considered IB registers).

3.2 I/O Control Registers

There are three I/O control registers (CONFADD, CONFDATA, and TRC) and these registers are all located in the CPU I/O space.

3.2.1 CONFADD—CONFIGURATION ADDRESS REGISTER

IO Address: 0CF8h
Default Value: 00000000h
Attribute: Read/Write
Size: 32 Bits

The CONFADD Register contains the address information for the next PCI configuration space access. Once the address is programmed into this register, the CPU can access the selected device register by a read/write to the CONFDATA Register. Only dword accesses are permitted to this register.

Bit	Description
31	**PCI Configuration Space Enable (CONFE):** This bit enables/disables access to the PCI configuration space. When CONFE = 1, PCI configuration space access is enabled. When CONFE = 0 (default), PCI configuration space access is disabled. When disabled, accesses to the CONFDATA Register, if not claimed on the PCI Bus, are forwarded to the ISA Bus.
30:24	**Reserved**
23:16	**Bus Number (BUSNUM):** This field selects the PCI Bus to be accessed. The PCI Bus behind the PSC is bus number 0 (00h). Thus, this field must be 00h for access to the PCIset's configuration registers.
15:11	**Device Number (DEVNUM):** This field selects the PCI Bus device to be accessed. The PSC uses this field to drive the IDSEL lines that select a specific PCI device during initialization. The IDSEL lines are only driven when BUSNUM = 00h. Otherwise, the PSC sends the configuration to a PCI-to-PCI bridge device. This field must be 05h for access to the PCIset's configuration registers (which is equivalent to IDSEL 16). Note that other PCI devices cannot use IDSEL 16.
10:8	**Function Number (FUNCNUM):** A device connected to the PCI Bus can have up to eight functions. This field selects a particular function within a device and must be 000 for access to the PCIset's configuration registers.
7:2	**Register Number (REGNUM):** This field is the configuration register offset address and indexes a dword in configuration space. REGNUM is used during initialization to select a specific device configuration registers.
1:0	**Reserved:** Fixed at 00.

3.2.2 CONFDATA—CONFIGURATION DATA REGISTER

IO Address: 0CFCh
Default Value: 00000000h
Attribute: Read/Write
Size: 32 Bits

The CONFDATA Register contains the data that is sent or received during a PCI configuration space access. Note that a read or write to this register accesses the PCI configuration space location specified by the contents of the CONFADD Register. CONFDATA supports CPU byte, word, and dword accesses.

3.2.3 TRC—TURBO/RESET CONTROL REGISTER

IO Address: 0CF9h
Default Value: 00h
Attribute: Read/Write
Size: 8 Bits

The TRC Register provides a means of generating soft or hard resets. During a hard reset, CPURST, PCIRST#, and RSTDRV are asserted for approximately 1 ms. A hard reset is initiated when this register is programmed for a hard reset or PWROK is asserted. During a soft reset, SRESET is asserted for a minimum of 16 Host Bus clocks. This register also selects the CPU De-Turbo mode. The TRC Register can only be accessed by the CPU with 8 bit IN or OUT instructions. Note that it is illegal for a PCI master or an ISA master to access the TRC Register.

Bit	Description
7:3	**Reserved:** Must be 0 when programming this register.
2	**Reset CPU (RCPU):** RCPU is used to initiate a hard reset or soft reset to the CPU, depending on the state of bit 1 of this register. Bit 1 must be set up prior to writing a 1 to bit 2. Thus, two write operations are required to initiate a reset. The first write programs bit 1 to the appropriate state while setting this bit to 0. The second write operation keeps bit 1 at its programmed state while setting this bit to a 1. When RCPU transitions from a 0 to a 1, a hard reset is initiated if bit 1 = 1 and a soft reset is initiated if bit 1 = 0.
1	**Reset CPU Mode (RCPUM):** This bit is used in conjunction with bit 2 of this register to initiate either a hard or soft reset. When RCPUM = 1, the PSC initiates a hard reset to the CPU when bit 2 transitions from 0 to 1. When RCPUM = 0, the PSC initiates a soft reset when bit 2 transitions from a 0 to a 1.
0	**Deturbo Mode (DM):** This bit enables/disables deturbo mode. When DM = 1, the 82420EX PCIset is in deturbo mode. In this mode, the PSC periodically asserts HOLD. The frequency of the HOLD assertion is fixed at once in 1024 Host Clocks. The duty cycle of the HOLD active period is controlled by the Deturbo Frequency Control (DFC) Register. When DM = 0, the Deturbo mode is disabled. Note that the deturbo counter does not start until HLDA is returned by the CPU. Deturbo mode can be used to maintain backward compatibility with older software packages that rely on the operating speed of the older processors. For accurate speed emulation, L1 caching should be disabled. If L1 is disabled during runtime, the following steps should be performed to make sure that all dirty data has been flushed from the cache to main memory before entering deturbo mode. Disable the L1 cache via the L1EN bit in the HOSTSEL Register. This prevents the KEN# signal from being asserted, which effectively disables the L1 cache. Thus, no new L1 cache line fills will occur. At this point, software executes the **WBINVD** of **INVD** instruction to flush the L1 cache, and then sets DM to 1. When exiting the deturbo mode, the system software must first set DM to 0, then enable L1 caching by writing to the HOSTSEL Register.

3.3 PCI Configuration Registers

This section describes the PCI configuration registers of the 82420 PCIset. The registers are listed in the order that they appear in Table 6.

3.3.1 VID—VENDOR IDENTIFICATION REGISTER

Address Offset: 00–01h
Default Value: 8086h
Attribute: Read Only
Size: 16 Bits

This 16-bit register, combined with the Device Identification Register, uniquely identifies any PCI device. The VID Register contains the vendor identification number assigned to Intel. Writes to this register have no effect.

3.3.2 DID—DEVICE IDENTIFICATION REGISTER

Address Offset: 02–03h
Default Value: 0486h
Attribute: Read Only
Size: 16 Bits

This 16-bit register, combined with the Vendor Identification register, uniquely identifies any PCI device. The 16-bit value in this register is the device number assigned to the 82425EX PSC. Writes to this register have no effect.

3.3.3 PCICOM—PCI COMMAND REGISTER

Address Offset: 04–05h
Default Value: 0007h
Attribute: Read/Write
Size: 16 Bits

This 16-bit register enables/disables the SERR# signal.

Bit	Description
15:9	**Reserved**
8	**SERR# Enable (SERRE):** SERRE enables/disables the SERR# signal. When SERRE = 1, SERR# is asserted if the PCIset detects a parity error during a main memory read cycle or a target abort is received on a PSC-initiated PCI cycle. When SERRE = 0, SERR# is never asserted.
7:0	**Not Used:** Defaults to 07h for compatibility reasons.

3.3.4 DS—DEVICE STATUS REGISTER

Address Offset: 06–07h
Default Value: 0200h
Attribute: Read Only and Read/Write Clear
Size: 16 Bits

DS is a 16-bit status register that reports the occurrence of a PCI master abort, PCI target abort, and main memory or cache parity errors. PCISTS also indicates the DEVSEL# timing that has been set by the PSC hardware.

Bit	Attribute	Description
15	R/WC	**Main Memory Parity Error (MMPERR):** When the PSC detects a parity error, this bit is set to 1. Software sets this bit to 0 by writing a 1 to it.
14	R/WC	**SERR# Status (SERRS):** When the PSC asserts the SERR# signal, this bit is set to a 1. Note that the SERR# signal is enabled/disabled in the PCICOM Register. When SERR# is enabled (via the PCICOM Register) and the PSC detects a parity error on a main memory read cycle or receives a target abort during a PSC-initiated PCI cycle, the PSC sets this bit to a 1.
13	R/WC	**Master Abort Status (MAS):** When a PSC-initiated PCI configuration cycle is not claimed, the PSC master aborts the cycle and sets this bit to a 1. For a CPU read, the PSC returns all 1s. When a memory cycle above 16 MBytes and not in an enabled BIOS region is not claimed, the PSC master-aborts the cycle and sets this bit to a 1. Software sets this bit to 0 by writing a 1 to it. **NOTE:** When a PSC-initiated PCI memory access under 16 MBytes or in an enabled BIOS range above 16 MBytes is not claimed, the PSC master aborts the cycle and forwards the cycle to ISA. When a PSC-initiated PCI I/O access is not claimed, the PSC master aborts the cycle and forwards the cycle to ISA. For these master aborts, the MAS bit is not set to 1.
12	R/WC	**Received Target Abort Status (RTAS):** When a PSC-initiated PCI transaction is terminated with a target abort, RTAS is set to 1. In addition, when the SERRE bit in the PCICOM Register is set to a 1 (enabling SERR#), the PSC asserts the SERR# signal. Software sets this bit to 0 by writing a 1 to it. Note that if the target aborted cycle is an I/O or memory read, the PSC completes the CPU cycle by returning RDY#.
11		**Reserved**
10:9	RO	**DEVSEL# Timing (DEVT):** This 2-bit field indicates the timing of the DEVSEL# signal when the PSC responds as a target. The PCI specification defines three allowable timings for assertion of DEVSEL#: 00 = fast, 01 = medium, and 10 = slow (DEVT = 11 is reserved). DEVT indicates the slowest time that a device asserts DEVSEL# for any bus command, except configuration read and write cycles. The 82420EX PCIset implements medium speed DEVSEL# timing and, therefore, bits[10:9] = 01 when read.
8:0		**Reserved**

3.3.5 RID—REVISION IDENTIFICATION REGISTER

Address Offset: 08h
Default Value: 00h
Attribute: Read Only
Size: 8 Bits

This register contains the revision number of the PSC. The Fabrication House ID Number and Revision Number correspond to bits 7-5 and the lower nibble respectively of the Revision Identification Register as follows:

bits 7-5 (upper 3 bits) Fabrication House ID Number
bits 3-0 (lower nibble) Revision Number

3.3.6 PCICON—PCI CONTROL REGISTER

Address Offset: 40h
Default Value: 00h
Attribute: Read /Write
Size: 8 Bits

The PCICON register enables/disables target abort and main memory DRAM parity error reporting. This register also selects the subtractive decode sample point, enables/disables PCI write buffers, and controls PCI bursting of consecutive CPU-to-PCI write cycles and byte merging.

Bit	Description
7	**Reserved**
6	**Target Abort Error Enable (TAEE):** When ATEE = 1 and a PSC-initiated cycle is target aborted, the PSC asserts SERR# for a PCI Clock. When TAEE = 0 (default) and a PSC-initiated cycle is target aborted, the PSC does not assert SERR#.
5	**DRAM Parity Error Enable (DPEE):** When DPEE = 1 and a main memory parity error is detected, the PSC asserts SERR# for a PCI Clock. When DPEE = 0 (default) and a main memory parity error is detected, the PSC will not assert SERR#.
4:3	**Subtractive Decode Sample Point (SDSP):** The SDSP field determines the DEVSEL# sample point, after which an inactive DEVSEL# results in the PSC forwarding the unclaimed PCI cycle to the ISA Bus (subtractive decoding). This setting should match the slowest device in the system. The values for this field and associated sampling point meaning are shown below. **Bits[4:3] — Sampling Point** 00 — Slow (default) 01 — Typical 10 — Fast 11 — Reserved
2	**PCI Posted Write Buffer Enable (PPBE):** When PPBE = 1, the PCI posted write buffers are enabled. When PPBE = 0 (default), the PCI posted write buffers are disabled.

Bit	Description
1	**CPU-to-PCI Bursting Enable (CPBE):** This bit enables/disables PCI burst cycles for CPU-to-PCI write cycles. When CPBE = 1, back-to-back sequential CPU memory writes are sent out on the PCI Bus as a burst cycle. When CPBE = 0 (default), CPU write cycles are always sent out on the PCI Bus as separate PCI memory write cycles.
0	**CPU-to-PCI Byte Merging (CPME):** Byte merging permits the PSC to merge the data of consecutive CPU-to-PCI byte/word writes within the same dword address, into the same posted write buffer location. The merged collection of bytes is then sent over the PCI Bus as a single dword. Byte merging is performed in the compatible VGA range only (0A0000–0BFFFFh). When CPME = 1, back-to-back CPU memory byte/word write cycles within the same dword address (in the 0A0000–0BFFFFh range) are merged into a single posted write buffer location. When CPME = 0 (default), or when the address location is outside of the VGA range, each memory write is stored in a separate posted write buffer location and sent separately over the PCI Bus. Some PCI graphics cards memory map their I/O location in the A0000h to B0000h memory region. If consecutive, multiple 8 or 16 bit write cycles are made to the add-in card at a memory mapped I/O location between A0000h–BFFFFh, the PSC will merge the data if the PSC is programmed for byte merging. The first write cycle will be written to the add-in card. However, subsequent write cycles will be overwritten in the PSC and never reach the add-in card. Because the consecutive, multiple write cycles are to the same address, the PSC will ‘‘merge’’ (overwrite the previous data) as long as the PCI bus is unavailable, causing the add-in card to not receive all the intended write cycles. Byte merging should be disabled when used with graphics cards that memory map I/O locations in the compatibility Video buffer area (A0000h–BFFFFh). Byte merging enhances graphics performances when used in operating systems that write to the video memory area in 8- or 16-bit writes (e.g. DOS). For operating systems that write to the video memory area in 32-bit writes, byte merging is not necessary.

3.3.7 HOSTDEV—HOST DEVICE CONTROL REGISTER

Address Offset: 44h
Default Value: 00h
Attribute: Read /Write
Size: 8 Bits

The HOSTDEV Register indicates to the PSC if there is a slave device, other than the PSC, that resides on the Host Bus. If there is another slave device present, the PSC sampling points for HDEV# and HRDY# are set in this register.

Bit	Description
7:3	**Reserved**
2	**Host Device Present (HDEVP):** When HDEVP = 1, there is a Host Bus slave device present. This device can claim any I/O or memory range that is not positively decoded by the PSC by asserting HDEV#. When HDEVP = 0 (default), there is no host bus slave device in the system.
1	**HDEV# Signal Sampling Point (HDEVSP):** HDEVSP specifies the maximum delay for HDEV# response and this bit only has meaning when HDEVP = 1. When HDEVSP = 1 (and HDEVP = 1), the PSC assumes that the Host Bus slave device asserts HDEV# with Host Bus fast timing (i.e., HDEV# can be asserted as late as one host clock after ADS# is asserted). In this case, the PSC samples HDEV# in the host cycle after ADS# and, if not asserted, forwards the cycle to the PCI Bus. When HDEVSP = 0 (and HDEVSP = 1), the PSC assumes that the Host Bus slave device asserts HDEV# with Host Bus slow timing (i.e., HDEV# can be asserted as late as two host clocks after ADS# is asserted). In this case, the PSC samples HDEV# in the host cycle after ADS# and one host cycle later and if not asserted, forwards the cycle to the PCI Bus.
0	**HRDY# Maximum Signal Sampling Point (HRDYSP):** HDEVSP specifies the delay from HRDY# to RDY# and this bit only has meaning when HDEVP = 1. When HRDYSP = 1 (and HDEVP = 1), the PSC assumes that the Host Bus slave device asserts HRDY# with Host Bus fast timing (i.e., Host RDY# is asserted in the same host clock as HRDY# is asserted). When HRDYSP = 0 (and HDEVP = 1), the PSC assumes that the Host Bus slave device asserts HRDY# with host bus slow timing (i.e., Host RDY# is asserted one host clock after HRDY# is asserted). Note that, when HDEVP = 0, HRDYSP has no meaning.

3.3.8 LBIDE—PCI LOCAL BUS IDE CONTROL REGISTER

Address Offset: 48–49h
Default Value: 0000h
Attribute: Read /Write
Size: 16 Bits

The LBIDE Register controls the PSC's IDE interface. The register determines when the PCI Local Bus IDE path will be used and the timing characteristics of the PCI Local Bus IDE cycle.

<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>15:13</td><td>Reserved</td></tr>
<tr><td>12:10</td><td>Recover Time (RCT[2:0]): This field controls the minimum time from the time IORDY is sampled asserted on the first cycle to the IOx# assertion of the next cycle. This recovery time mechanism applies to all cycles using the fast timing bank, even if the next cycle is a compatible cycle. For example, if a first cycle is a data port access using the fast timing bank and the next cycle is to a control or status port, LBIDE# is negated and the full setup protocol occurs prior to the second cycle. Normally, this setup protocol is longer than the programmed recovery time. However, if the setup protocol is shorter, the proper recovery time must still be met. The value of this field determines the minimum number of PCI clocks between the last IORDY sample point and the IOx# strobe of the next cycle.

<table>
<tr><th>RCT[2:0]
Bits[12:10]</th><th>Recovery Time</th><th>RCT[2:0]
Bits[12:10]</th><th>Recovery Time</th></tr>
<tr><td>000</td><td>8</td><td>100</td><td>4</td></tr>
<tr><td>001</td><td>7</td><td>101</td><td>3</td></tr>
<tr><td>010</td><td>6</td><td>110</td><td>Note</td></tr>
<tr><td>011</td><td>5</td><td>111</td><td>Note</td></tr>
</table>

NOTE: The recovery time is 3 PCI clocks.</td></tr>
<tr><td>9:8</td><td>IORDY Sample Point (ISP[1:0]): This field determines the number of clocks between IOx# assertion and the first IORDY sample point (see the following table). IORDY is sampled for the first time on the programmed number of clocks (number of low-to-high clock transitions) following the assertion of IOx#. If IORDY is negated when sampled, wait-states are inserted until IORDY is sampled asserted.

<table>
<tr><th>ISP[1:0]
Bits[9:8]</th><th>IORDY Sampling Point</th></tr>
<tr><td>00</td><td>6 Clocks (default)</td></tr>
<tr><td>01</td><td>5 Clocks</td></tr>
<tr><td>10</td><td>4 Clocks</td></tr>
<tr><td>11</td><td>3 Clocks</td></tr>
</table></td></tr>
<tr><td>7:6</td><td>Reserved</td></tr>
<tr><td>5:4</td><td>IORDY Sample Point Enable Drive Select (ISPEDS[1:0]): ISPEDS[1:0] enable/disable the sampling of IORDY for drive 0 and 1. When this feature is enabled for a drive (via this field), and the drive is selected (via a copy of bit 4 of 1x6h), all accesses to the enabled I/O address range sample IORDY. The IORDY sample point is specified by the ISP[1:0] field. When the drive is disabled (0), IORDY sampling is disabled. The internal IORDY signal is forced asserted guaran-teeing that IORDY is sampled asserted at the first sample point as specified by the ISP[1:0] field.

<table>
<tr><th>ISPEDS[1:0]
Bits[5:4]</th><th>IORDY Sampling Point Enable Drive Select</th></tr>
<tr><td>00</td><td>Neither Drive Enabled (default)</td></tr>
<tr><td>01</td><td>Drive 0 Enabled</td></tr>
<tr><td>10</td><td>Drive 1 Enabled</td></tr>
<tr><td>11</td><td>Both Drives Enabled</td></tr>
</table></td></tr>
</table>

Bit	Description
3:2	**Fast Timing Bank Drive Select 1 (FTBDS1):** FTBDS[1:0] enable/disable the fast timing PCI local bus IDE path for drive 0 and 1. When this feature is enabled for a drive (via this field), and the drive is selected (via a copy of bit 4 of 1x6h), all accesses to the enabled I/O address range will use the fast timing bank PCI local bus IDE path. Note that accesses to all non-data ports of the enabled I/O address range use the 8-bit compatible timing PCI local bus path. When the drive is disabled (0), accesses to the data port of the selected I/O address range use the 16-bit compatible timing PCI local bus path. **Bits[3:2] — Fast Timing Bank Select** 00 — Neither Drive Enabled (default) 01 — Drive 0 Enabled 10 — Drive 1 Enabled 11 — Both Drives Enabled
1:0	**Primary/Secondary PCI IDE Enable (IDEE):** This field enables/disables the PCI IDE, and, if enabled, selects the primary/secondary IDE address that is used as shown below. Accesses to the unselected address range (primary/secondary) are forwarded to the ISA Bus. **Bits[1:0] — Primary/Secondary IDE Address Select** 00 — IDE Disabled (default) 01 — Primary: 1F0–1F7h, 3F6h,3F7h 10 — Secondary: 170–177H, 376h, 377h 11 — Reserved

LBIDE Programming Information

The BIOS code will assess the CPU frequency and drive capabilities, and then program the timing fields appropriately. Table 8 shows the typical settings of the various cycle timing bits for the supported CPU frequencies and IDE modes. The table assumes that the drives are utilizing IORDY. If IORDY is not utilized, additional wait-states may be deleted via the ISP bits.

Table 8. Typical Register Settings for Different CPU Frequencies

PCI Frequency	IDE Mode	RCT(2:0)		ISP(1:0)		Cycle Length (ns) 1x Clock Mode(1)
		Bits[12:10]	Clocks	Bits[9:8]	Clocks	
20	1	100	4	10	4	400(1)
25	1	101	6	10	4	400
33	1	001	7	00	6	390
20	2	110	2	11	3	300(2)
25	2	110	2	10	4	280(2)
33	2	101	3	01	5	240
20	3	110	2	11	3	300(2)
25	3	110	2	11	3	240(2)
33	3	101	3	10	4	210

NOTES:
1. The clock modes are determined by strapping options at powerup and the mode is reflected in the HOSTSEL Register.
2. Cycle times are governed by the inherent RDY#, ADS#, and address decoding delay between back-to-back cycles rather than the programmed value in the RCT field.

3.3.9 IORT—ISA I/O RECOVERY TIMER REGISTER

Address Offset: 4Ch
Default Value: 4Dh
Attribute: Read/Write
Size: 8 bits

The IORT Register provides ISA I/O recovery time control and enables/disables the aliasing of addresses 80–8Fh and 90–9Fh. The I/O recovery mechanism in the IB adds recovery delay between CPU or PCI master originated 8-bit and 16-bit I/O cycles to the ISA Bus. The IB automatically forces a minimum delay of 3.5 SYSCLKs between back-to-back 8 and 16 bit I/O cycles to the ISA Bus. This delay is measured from the rising edge of the I/O command (IOR# or IOW#) to the falling edge of the next I/O command. If a delay of greater than 3.5 SYSCLKs is required, the ISA I/O Recovery Time Register can be programmed to increase the delay in increments of SYSCLKs. No additional delay is inserted for back-to-back I/O ''sub-cycles'' generated as a result of byte assembly or disassembly. This register defaults to 8- and 16-bit recovery enabled with one SYSCLK clock added to the standard I/O recovery.

<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>7</td><td>DMA Reserved Page Register Aliasing Disable (DMAAD): When DMAAD = 0 (default), the IB aliases I/O accesses in the 90–9Fh range to the 80–8Fh range. In this case, the IB only forwards CPU/PCI write accesses to the 90–9Fh range to the ISA Bus. When DMAAD = 1, the IB forwards both CPU/PCI read and write accesses to these address locations to the ISA Bus (i.e. the I/O address locations are no longer considered IB internal registers). Note, that port 92h is always a distinct ISA register in the 90–9Fh range and is always forwarded to the ISA Bus. When DMAAD = 1, ISA master accesses to the 90–9Fh range are ignored by the IB.

When this bit is set to 1, the IB does not re-load the power management Fast Off-Timer with its original value when accesses to the 90–9Fh address range are decoded.</td></tr>
<tr><td>6</td><td>8-Bit I/O Recovery Select (IOR8E): When IOR8E = 1, bits[5:3] select the I/O recovery time. When IOR8E = 0, programmable recovery times are disabled and the standard recovery time of 3.5 SYSCLKS is inserted.</td></tr>
<tr><td>5:3</td><td>8-Bit I/O Recovery Times (IOR8): This 3-bit field defines the recovery time for 8-bit I/O as shown Below. Programmable delays between back-to-back 8-bit PCI or CPU cycles to ISA I/O slaves are shown in terms of additional ISA clock recovery cycles (SYSCLK). The selected delay programmed into this field is enabled/disabled via bit 6 of this register.

<table>
<tr><th>Bits[5:3]</th><th>8-Bit I/O Recover Time</th><th>Bits[5:3]</th><th>8-Bit I/O Recover Time</th></tr>
<tr><td>000</td><td>8 SYSCLKs</td><td>100</td><td>4 SYSCLKs</td></tr>
<tr><td>001</td><td>1 SYSCLKs (default)</td><td>101</td><td>5 SYSCLKs</td></tr>
<tr><td>010</td><td>2 SYSCLKs</td><td>110</td><td>6 SYSCLKs</td></tr>
<tr><td>011</td><td>3 SYSCLKs</td><td>111</td><td>7 SYSCLKs</td></tr>
</table></td></tr>
<tr><td>2</td><td>16-Bit I/O Recovery Enable (IOR16E): When IOR16E = 1, bits[1:0] select the I/O recovery time. When IOR16E = 0, programmable recovery times are disabled and the standard recovery time of 3.5 SYSCLKS is inserted.</td></tr>
</table>

Bit	Description
1:0	**16-Bit I/O Recovery Times (IOR16):** This 2-bit field defines the recovery time for 16-bit I/O as shown below. Programmable delays between back-to-back 16-bit PCI or CPU cycles to ISA I/O slaves are shown in terms of additional ISA clock recovery cycles (SYSCLK). The selected delay programmed into this field is enabled/disabled via bit 2 of this register. **Bits[1:0] — 16-Bit I/O Recover Time** 00 — 4 SYSCLKs 01 — 1 SYSCLKs (default) 10 — 2 SYSCLKs 11 — 3 SYSCLKs

3.3.10 PREV—PART REVISION REGISTER

Address Offset: 4Dh
Default Value: 00h
Attribute: Read/Write. Read Only
Size: 8 bits

This register provides the device stepping information for the IB and enables/disables DMA and ISA master accesses to DRAM BIOS locations E0000–EFFFFh. Bits 0 and 1 in this register are hardwired and write accesses have no effect.

Bit	Attribute	Description
7:5		**IB Fabrication House ID**
4	R/W	**E0000–EFFFFh ISA Forwarding Enable:** When bit 4 = 1 (and bit 6 in the XBCSA Register is set to 0), ISA master and DMA accesses to memory locations E0000–EFFFFh are forwarded to main memory. When bit 4 = 0 (default), ISA master and DMA accesses to this memory region are confined to the ISA Bus. Note that if bit 6 = 1 in the XBCSA Register, memory accesses to memory locations E0000–EFFFFh are always confined to the ISA Bus, regardless of the setting of bit 4 of this register.
3:0	RO	**Revision ID:** This field contains the device stepping information for the 82426EX IB.

3.3.11 XBCSA—X-BUS CHIP SELECT A REGISTER

Address Offset: 4Eh
Default Value: 03h
Attribute: Read/Write
Size: 8 bits

This register enables/disables accesses to the real time clock (RTC), keyboard controller (KBC), and BIOS. Disabling any of these bits prevents the chip select and X-Bus output enable control signal (XBUSOE#) for that device from being generated. The XBCSA Register also provides coprocessor error and mouse functions.

Bit	Description
7	**Extended BIOS Enable (EXBIOSE):** When EXBIOSE = 1 (enabled), CPU or PCI master accesses to locations FFF80000–FFFDFFFFh, that are not claimed by a PCI device are forwarded to the ISA Bus and results in the generation of BIOSCS# and XBUSOE#. Note that forwarding this region at the top of 4 GBytes to the ISA Bus (24-bit addressing) results in aliasing this 384 KByte region to the top of the 16 MByte ISA memory space. To avoid contention, ISA add-in memory must not be present in this space. When EXBIOSE = 0 (disabled: default), BIOSCS# or XBUSOE# are not generated. CPU accesses to locations FFF80000–FFFDFFFFh, that are not claimed by PCI devices are master-aborted by the PSC. Note that the Master Abort Status bit is set to 1 (DS Register). For reads, data of all 1's is returned to the CPU. PCI master accesses to locations FFF80000–FFFDFFFFh that are not claimed by PCI devices are ignored by the PSC.
6	**Lower BIOS Enable (LBIOSE):** When LBIOSE = 1 (enabled: default), CPU, PCI master, or ISA master accesses to the lower 64 KByte BIOS block at the top of 1 MByte (E0000–EFFFFh), or the aliases at the top of 4 GByte, that are not claimed by PCI devices, result in the generation of BIOSCS# and XBUSOE#. Note that forwarding this region at the top of 4 GBytes to the ISA Bus (24-bit addressing) results in aliasing this region to the top of the 16 MByte ISA memory space. To avoid contention, ISA add-in memory must not be present in this space. When LBIOSE = 0 (disabled: default), BIOSCS# or XBUSOE# are not generated during these accesses. Also, when LBIOSE = 0 (and bit 4 is 1 in the IB's PREV Register), ISA master or DMA accesses to this region are forwarded to main memory. CPU accesses to the lower 64 KByte BIOS region (0FFFE0000–0FFFEFFFFh) that are not claimed by PCI devices are master-aborted by the PSC. The Master Abort Status bit is set to 1 (DS Register). For reads, data of all 1's is returned to the CPU. PCI master accesses to the lower 64 KByte BIOS region (0FFFE0000–0FFFEFFFFh), that are not claimed by PCI devices, are ignored by the PSC.
5	**Coprocessor Error Function Enable (CPEE):** This bit enables/disables the coprocessor error support. When CPEE = 1 (enabled), the FERR# input, when asserted, triggers IRQ13 (internal). FERR# also gates the IGNNE# output. This bit defaults to disabled (0).
4	**IRQ12/M Mouse Function Enable (IRQ12/ME):** When bit 4 = 1, IRQ12/M provides the mouse function. When bit 4 = 0 (default), IRQ12/M provides the standard IRQ12 interrupt function.
3	**Reserved:** Must be set to 0.
2	**BIOS Memory Write Protect (BIOSWP):** When BIOSWP = 1 (read/write access), BIOSCS# is asserted for BIOS memory read and write cycles in the decoded BIOS region. When BIOSWP = 0 (write protect: default), BIOSCS# is only asserted for BIOS read cycles.
1	**Keyboard Controller Address Enable (KBCAE):** When KBCAE = 1 (enabled: default), the keyboard controller chip select signal (KBCCS#) and the XBUSOE# signals are generated for accesses to the keyboard controller address locations 60h, 62h, 64h, and 66h. When KBCAE = 0 (disabled), KBCCS# and XBUSOE# are not generated for these accesses.
0	**RTC Address Enable (RTCAE):** When RTCAE = 1 (enabled: default), the RTCCS#, RTCALE, and XBUSOE# signals are generated for accesses to the RTC address locations 70–77h. When RTCAE = 0 (disable), the RTCCS#, RTCALE, and XBUSOE# signals are not generated for accesses to these addresses.

3.3.12 HOSTSEL—HOST SELECT REGISTER

Address Offset: 50h
Default Value: 00000xx0 (x = Depends on hardware strapping options)
Attribute: Read/Write
Size: 8 Bits

The HOSTSEL Register enables/disables the L1 cache, indicates the clock configuration selected by hardware strapping options, and selects the L1 caching policy.

Bit	Description
7:4	**Reserved**
3	**L1 Caching Policy Select (L1CPSEL):** L1CPSEL selects the caching policy for the L1 cache. When L1CPSEL = 1, the L1 caching policy is write-back and when L1CPSEL = 0, the caching policy is write-through.
2:1	**Clock Configuration Status (CLKCONFS):** Clock configuration is determined by hardware strapping options on the SIDLE# and CMDV# signal pins at power-up (see Section 4.15, Clocks). This field is read only. CLKCONFS indicates the clock mode and frequencies selected by the strapping options. **Bits[2:1] / HCLK / PCICLK / Clock Mode** 00 — 25 MHz — 25 MHz — 1x 01 — 33 MHz — 33 MHz — 1x 10 — Reserved 11 — Reserved
0	**L1 Cache Enable (L1CE):** L1CE enables/disables the first level cache in the CPU. When L1CE = 1 (enable), the PSC responds to the CPU with KEN# asserted for cacheable memory cycles. When L1CE = 0 (disable), the KEN# signal is always negated to the CPU. This prevents new cache line fills to either the first level or second level caches.

3.3.13 DFC—DETURBO FREQUENCY CONTROL REGISTER

Address Offset: 51h
Default Value: 80h
Attribute: Read /Write
Size: 8 Bits

Some old software packages that rely on the operating speed of the processor do not work on today's faster systems. To maintain backward compatibility with these software packages, the 82420EX PCIset provides a mechanism to emulate the operating speed of PC/AT systems. This emulation is achieved with the deturbo mode (enabled/disabled via the Turbo/Reset Control Register). When the deturbo mode is enabled, the PSC periodically asserts the HOLD signal to slow down the effective speed of the CPU. The frequency of the HOLD assertion is fixed to once in 1024 Host Clocks. The duty cycle of the HOLD active period is controlled by the DFC Register.

Bit	Description
7:0	**Deturbo Mode Frequency Adjustment Value:** This 8-bit value effectively defines the duty cycle of the HOLD signal. The value programmed into this register is compared against a free running 8-bit counter running at 1/4 the CPU clock. When the counter is greater than the value specified in this register, HOLD is asserted to the CPU. HOLD is negated when the counter value is equal to or smaller than the contents of this register. HOLD is negated when the counter rolls over to 00h. Note that the deturbo counter does not start until HLDA is returned by the CPU. The deturbo emulation speed is directly proportional to the value in this register. The smaller the value in this register the lower the deturbo emulation speed.

3.3.14 SCC—SECONDARY (L2) CACHE CONTROL REGISTER

Address Offset: 52–53h
Default Value: 0000h
Attribute: Read /Write
Size: 16 Bits

This 16-bit register defines the L2 cache operations. SCC enables/disables the L2 cache, adjusts cache size, selects the cache write policy, defines the cache SRAM type, and selects various read/write cache cycle times. In addition, a cache miss can be forced for each access permitting software to initialize the cache. Cache hits can also be forced permitting software to determine the size of the L2 cache memory.

NOTE:
The L2 timings must be programmed at least as fast as the DRAM timings.

Bit	Description
15:13	**Reserved**
12	**Hit Dirty Write Cycle Timing (HDWRTIME):** When HDWRTIME = 1, the PSC performs 0 wait-state accesses (2-1-1-1) for hit dirty write cycles. When HDWRTIME = 0 (default), the access time for write hit dirty cycles is determined by WRTIME (bit 11). See Section 3.3.14.1.
11	**Write Cycle Timing (WRTIME):** When WRTIME = 1, the timing for PSC L2 cache write accesses is 3-2-2-2. In this case, the WRTIME bit is ignored for write hit dirty cycles. When WRTIME = 0 (default), the timing for PSC L2 cache write accesses is 4-2-2-2. See Section 3.3.14.1.
10:9	**Subsequent Read Timing (SUBRD):** This field determines the access time for subsequent reads to the L2 cache as shown below. **Bits[10:9] — Subsequent Cache Read Timing** 00 — X-3-3-3 (default) 01 — X-2-2-2 10 — X-1-1-1 11 — Reserved
8	**Initial Read Timing (INITRD):** This bit determines the access time for initial reads to the L2 cache. When INITRD = 1, the initial read timing is 2-X-X-X. When INITRD = 0 (default), the initial read timing is 3-X-X-X.

Bit	Description
7	**Reserved**
6	**Cache Force Hit (L2FHIT):** When L2FHIT = 0, the cache operation is normal. When L2FHIT = 1, the L2 data SRAMs are accessed, for cacheable data reads and writes, as if they were main memory. Since all data reads and writes hit the cache, none of the data cycles go to main memory. Thus, BIOS can determine the L2 cache size and configuration during POST. While L2FHIT = 1 forces a hit for cacheable data cycles, all code reads are forced to be non-cacheable. Thus, the CPU can perform code read cycles from main memory without the L2 cache generating write-back cycles and without the L2 generating code-read-allocate cycles (that may interfere with the L2 sizing algorithm). When in L2FHIT mode, the primary (L1) cache must be disabled. L2 configuration is determined by setting the cache in Interleaved mode and performing line write/read. The L2 cache size is determined by setting L2SIZE (from 64 KBytes and up) and performing a write to location (K + cache-size) and a read to location (K). When L2SIZE = 000, the L2FHIT bit has no effect.
5	**L2 Cache Force Miss Clean (L2FMISS):** When L2FMISS = 0 (default), the L2 Cache operation is normal. When L2FMISS = 1, all cacheable accesses to L2 are forced to be a cache miss. This bit is used to initialize the cache with valid locations. BIOS can set this bit and read a block of main memory equal to the cache size. This fills the cache with valid data. Once the cache is initialized, software sets this bit to 0, and the PSC keeps the cache coherent with main memory. When L2SIZE = 000 (L2 disabled), the L2FMISS bit has no effect.
4	**Cache Configuration (L2CONF):** This bit determines the configuration of the L2 cache SRAMs. For an interleaved memory configuration, L2CONF = 1 and for a non-interleaved configuration, L2CONF = 0 (default).
3	**Cache Write Policy (L2WPOL):** This bit determines the L2 cache policy. When WRPOL = 1, the L2 cache policy is write-back. When WRPOL = 0 (default), the cache policy is write-through.
2:0	**Cache Size (L2SIZE):** This field determines the L2 cache size as shown below.

Bits[2:0]	Cache Size	Bits[2:0]	Cache Size
000	L2 Cache Disabled (default)	100	512 KBytes
001	64 KBytes	101	Reserved
010	128 KBytes	110	Reserved
011	256 KBytes	111	Reserved

3.3.14.1 L2 Write Timing

Bits 11 and 12 control the write timing for the L2 cache controller. Bit 12 = 1, bit 11 = 0 is an invalid combination which will cause the PSC to lock up. The following table shows the various bit 11, 12 combinations and how they program the L2 cache controller write timings.

Bit 12	Bit 11	L2 Cache Write Timing
0	0	4-2-2-2
0	1	3-2-2-2
1	0	Invalid
1	1	2-1-1-1 (hit dirty write cycles)

3.3.15 DRAMC—DRAM CONTROL REGISTER

Address Offset: 56–57h
Default Value: 0000h
Access: Read/Write
Size: 16 bits

The DRAMC Register selects various DRAM interface timing parameters. This register also enables/disables fast page mode for DRAM access, enables/disables CAS pipelining, and provides a refresh test option.

NOTE:
The L2 timings must be programmed at least as fast as the DRAM timings.

<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>15</td><td>Muxed Address Hold Time (MAH): This bit determines the number of clocks from RAS# or CAS# active before MA can be changed. When MAH is 1, the hold time is 0.5 active clocks and, when MAH is 0 (default), the hold time is 1.0 active clock.</td></tr>
<tr><td>14</td><td>Muxed Address Setup Time (MASU): This bit determines the number of clocks from Muxed Address driven to RAS# or CAS# active. When MASU is 1, the address setup is 0.5 active clocks and, when MASU is 0 (default), the address setup is 1.0 active clocks.</td></tr>
<tr><td>13</td><td>CAS Write Timing (CASWR): This bit determines the number of clocks CAS# remains active during a write access, and inactive between accesses as shown below.

Bit 13 — Active, Inactive Clocks
0 — 2 Active, 1 Inactive (default)
1 — 1 Active, 1 Inactive</td></tr>
<tr><td>12:11</td><td>CAS Read Timing (CASRD): This field determines the number of clocks CAS# remains active during a read access and inactive between accesses as shown below.

Bits[12:11] — Active, Inactive Clocks
00 — 3 Active, 1 Inactive (default)
01 — 2 Active, 1 Inactive
10 — 1.5 Active, 0.5 Inactive
11 — 1 Active, 1 Inactive</td></tr>
<tr><td>10:8</td><td>RAS Precharge Timing (RASPRE): This field determines the minimum number of Host Bus clocks that RAS# remains inactive as shown below.

Bits[10:8] — RAS Pre-Charge Time — Bits[10:8] — RAS Pre-Charge Time
000 — 4 Clocks (default) — 100 — 1.5 Clocks
001 — 3 Clocks — 101 — 1 Clock
010 — 2 Clocks — 110 — Reserved
011 — Reserved — 111 — Reserved</td></tr>
<tr><td>7:6</td><td>Reserved</td></tr>
<tr><td>5</td><td>Pipelined CAS Enable (PCASEN): When PCASEN = 1, the DRAM controller does not provide any time between CAS[3:0]# negation and CAS[7:4]# assertion (interleaved row only). When PCASEN = 0 (default), the DRAM controller provides 1 Host Bus clock between CAS[3:0]# negation and CAS[7:4]# assertion (interleaved row only).</td></tr>
</table>

Bit	Description
4	**Refresh Test Enable (REFTSTE):** When REFTST = 1, a test mode for the refresh generator is enabled. In this mode, a refresh request is generated every 32 HCLK cycles. When REFTST = 0 (default), the DRAM controller generates a refresh cycle every 15 μs.
3	**Fast Page Write Enable (FPWE):** This bit permits the PSC to keep the currently accessed DRAM page active following a CPU write cycle. When FPWE = 1, the PSC keeps the page open (keeps RAS# asserted) following a write cycle to main memory. When FPWE = 0 (default), the PSC closes the page (negates RAS#) following a write cycle to main memory, creating a row miss for every CPU write.
2	**Fast Page Data Read Enable (FPDRE):** This bit permits the PSC to keep the currently accessed DRAM page active following a CPU data read cycle. When FPDRE = 1, the PSC keeps the page open (keeps RAS# asserted) following a data read cycle to main memory. When FPDRE = 0 (default), the PSC closes the page (negates RAS#) following a data read cycle to main memory, creating a row miss for every CPU data read.
1	**Fast Page Code Read Enable (FPCRE):** This bit permits the PSC to keep the currently accessed DRAM page active following a CPU code read cycle. When FPCRE = 1, the PSC keeps the page open (keeps RAS# asserted) following a code read cycle to main memory. When FPCRE = 0 (default), the PSC closes the page (negates RAS#) following a code read cycle to main memory, creating a row miss for every CPU code read.
0	**Reserved**

3.3.16 PAM[6:0]—PROGRAMMABLE ATTRIBUTE MAP REGISTERS

Address Offset: PAM6(5Fh), PAM5(5Eh), PAM4(5Dh), PAM3(5Ch)
PAM2(5Bh), PAM1(5Ah) PAM0(59h)
Default Value: PAM[6:0] = 00h
Attribute: Read/Write

The 82420EX PCIset allows programmable memory and cacheability attributes on 13 memory segments of various sizes in the ISA compatibility hole—640 KByte to 1 MByte address range. Seven Programmable Attribute Map (PAM) Registers support these features. Four bits specify cacheability and memory attributes for each memory segment. These attributes are:

RE Read Enable. When RE = 1, the CPU read accesses to the corresponding memory segment are directed to main memory. Conversely, when RE = 0, the CPU read accesses are forwarded to PCI and, if not claimed on PCI, are forwarded to ISA.

WE Write Enable. When WE = 1, the PCI write accesses to the corresponding memory segment are directed to main memory. Conversely, when WE = 0, the CPU write accesses forwarded to PCI and, if not claimed on PCI, are forwarded to ISA.

CE Cache Enable. When CE = 1, the corresponding memory segment is cacheable. It is illegal to set CE = 1 and RE = 0 for the same segment. When CE = 1 and WE = 0, the corresponding memory range is not cached in the L1 cache (KEN# is negated on CPU accesses). However, it is cached and write protected in the L2 cache. The L2 cache handles cached write protected ranges as follows:

Code read (L2 miss): L2 line is allocated, data is read from main memory.
Data read (L2 miss): data is read from main memory.
Any read (L2 hit): data is read from the L2 cache
Any write: subtractively decoded to PCI Bus.

PE PCI Enable. When PE = 1, the corresponding memory range is accessible by PCI masters, as a function of the RE, WE and CE bits setting. When PE = 0, the corresponding memory range is inaccessible by PCI masters (the PCI master cycles are either claimed by PCI slaves or sent to ISA).

Each PAM Register controls two ranges as shown in Table 9.

NOTE:
The combination RE = 0 and CE = 1 is illegal.

Table 9. PAM Registers and Associated Memory Ranges

PAM Reg	Attribute Bits				Memory Segment	Comments	Offset
PAM0[3:0]						Reserved	59h
PAM0[7:4]	PE	CE	WE	RE	0F0000–0FFFFFh	BIOS Area	59h
PAM1[3:0]	PE	CE	WE	RE	0C0000–0C3FFFh	ISA Add-on BIOS	5Ah
PAM1[7:4]	PE	CE	WE	RE	0C4000–0C7FFFh	ISA Add-on BIOS	5Ah
PAM2[3:0]	PE	CE	WE	RE	0C8000–0CBFFFh	ISA Add-on BIOS	5Bh
PAM2[7:4]	PE	CE	WE	RE	0CC000–0CFFFFh	ISA Add-on BIOS	5Bh
PAM3[3:0]	PE	CE	WE	RE	0D0000–0D3FFFh	ISA Add-on BIOS	5Ch
PAM3[7:4]	PE	CE	WE	RE	0D4000–0D7FFFh	ISA Add-on BIOS	5Ch
PAM4[3:0]	PE	CE	WE	RE	0D8000–0DBFFFh	ISA Add-on BIOS	5Dh
PAM4[7:4]	PE	CE	WE	RE	0DC000–0DFFFFh	ISA Add-on BIOS	5Dh
PAM5[3:0]	PE	CE	WE	RE	0E0000–0E3FFFh	BIOS Extension	5Eh
PAM5[7:4]	PE	CE	WE	RE	0E4000–0E7FFFh	BIOS Extension	5Eh
PAM6[3:0]	PE	CE	WE	RE	0E8000–0EBFFFh	BIOS Extension	5Fh
PAM6[7:4]	PE	CE	WE	RE	0EC000–0EFFFFh	BIOS Extension	5Fh

DOS Application Area (00000h–9FFFFh)

The 640 KByte DOS application area always has read, write, and cacheability attributes enabled and are not programmable for the 0–640 KByte region.

Video Buffer Area (A0000h–BFFFFh)

This 128 KByte area is not controlled by attribute bits. It is always subtractively decoded to ISA.

Expansion Area (C0000h–DFFFFh)

This 128 KByte area is divided into eight 16 KByte segments. Each segment can be assigned one of four Read/Write states: read-only, write-only, read/write, or disabled (memory that is disabled is not remapped elsewhere). Cacheability status can also be specified for each segment.

Extended System BIOS Area (E0000h–EFFFFh)

This 64 KByte area is divided into four 16 KByte segments. Each segment can be assigned independent cacheability, read, and write attributes. Memory segments that are disabled are not remapped elsewhere.

System BIOS Area (F0000h–FFFFFh)

This area is a single 64 KByte segment. This segment can be assigned cacheability, read, and write attributes and PCI enabled.

Extended Memory Area (100000h–FFFFFFFFh)

The extended memory area can be split into several parts;

- BIOS area from 4 GByte to (4 GByte minus 512 KByte) (aliased on ISA at 16 MByte minus 15.5 MByte)
- Main memory from 1 MByte to a maximum of 128 MBytes
- PCI memory space from TOM to 128 MBytes or, (2 GBytes minus 128 MBytes) to (2 GByte plus 128 MByte), or 4 GByte to (4 GByte minus 128 MByte)

On power-up or reset the CPU vectors to the Flash BIOS area, mapped in the range of 4 GByte to 4 GByte–512 KByte. However, this area is physically mapped on ISA. Since these addresses are in the upper 4 GByte range, the request is directed to PCI. The 82420EX PCIset strips the upper address bits to effectively map the BIOS on ISA in the area between 16 MByte to 15.5 MByte.

The main memory space can occupy extended memory from a minimum of 1 MByte up to 128 MBytes. This memory is cacheable. The following areas may be occupied by PCI memory: the address space on PCI from TOM to 128 MBytes, between the Flash BIOS (4 GByte minus 512 KByte) and (4 GByte minus 128 MByte), and the range from (2 GBytes minus 128 MBytes) to (2 GByte plus 128 MByte) may be occupied by PCI memory. This memory space is not cacheable.

3.3.17 DRB—DRAM ROW BOUNDARY REGISTERS

Address Offset: DRB4(64h), DRB3(63h) DRB2(62h) DRB1(61h), DRB0(60h)
Default Value: 01h (for each DRB)
Access: Read/Write
Size: 8 bits

The PSC supports up to 5 rows of DRAM. When populated, each row contains 32 (non-interleaved) or 64 (interleaved) bits of data. The DRAM Row Boundary registers define upper and lower addresses for each DRAM row. Contents of these 8-bit registers represent the amount of memory in MBytes.

DRB0 = Total amount of memory in row 0 (in MBytes)
DRB1 = Total amount of memory in row 0 + row 1 (in MBytes)
DRB2 = Total amount of memory in row 0 + row 1 + row 2 (in MBytes)
DRB3 = Total amount of memory in row 0 + row 1 + row 2 + row 3 (in MBytes)
DRB4 = Total amount of memory in row 0 + row 1 + row 2 + row 3 + row4 (in MBytes)

The DRAM array can be configured with SIMMs that have address depths of 256 KByte, 1 MByte, and 4 MByte. Each register defines an address range that causes a particular RAS# line to assert (e.g. if the first DRAM row is 2 MBytes in size, then accesses within the 0 to 2 MByte range causes the RAS0# line to be asserted). The DRAM Row Boundary (DRB) Registers are programmed with an 8-bit upper address limit value.

Bit	Description
7:0	**Memory Boundary in MBytes:** This 8-bit value is used to determine the upper address limit of this row (i.e., this row - previous row = row size). Unpopulated rows have a value equal to the previous row (row size = 0). The value programmed into DRB4 always reflects the maximum amount of DRAM in the system.

Example 1:
If SIMM0 contains a 256K x 36 SIMM,(which is equivalent to 1 MByte DRAM), DRB0 is set to 01h. If this is the only SIMM in the system, DRB[4:1] are each set to 01h.

Example 2:
One way to achieve maximum main memory is to populate SIMMs 0-3 with 8M x 36 double-sided SIMMs (which have 32 MBytes each). In this case, DRB[4:0] would be programmed as follows: DRB0 = 20h, DRB1 = 40h, DRB2 = 60h, DRB3 = 80h, DRB4 = 80h.

3.3.18 PIRQ1RC/PIRQ0RC—PIRQ ROUTE CONTROL REGISTERS

Address Offset: 66h (PIRQ0RC)
67h (PIRQ1RC)
Default Value: PIRQ0RC 80h
PIRQ1RC 80h
Attribute: Read/Write
Size: 8 bits

The PIRQ1RC/PIRQ0RC Registers control the routing of PIRQ[1:0] signals to the internal IRQ inputs of the interrupt controller. Each PIRQx# can be independently routed to any one of 11 interrupts. One or both PIRQx# lines can be routed to the same IRQx input. Note that the IRQ selected through bits[3:0] must be set to level sensitive mode in the corresponding ELCR Register.

Bit	Description
7	**PIRQx Interrupt Signal Routing Enable:** When bit 7 = 0 (enabled), PIRQx# is routed to the IRQ selected by bits[3:0] of this register. When bit 7 = 1 (disabled: default), the PIRQx# signal is not routed to any IRQ line.
6:4	**Reserved:** Read as zeros.
3:0	**PIRQx Interrupt Signal Routing:** When bit 7 = 0, bits[3:0] select how each PIRQx# is routed to each internal 8259 IRQx. The routing for different values of this field are shown below.

Bits[3:0]	PIRQx Interrupt Routing	Bits[3:0]	PIRQx Interrupt Routing
0000	Reserved (default)	1000	Reserved
0001	Reserved	1001	IRQ9
0010	Reserved	1010	IRQ10
0011	IRQ3	1011	IRQ11
0100	IRQ4	1100	IRQ12
0101	IRQ5	1101	Reserved
0110	IRQ6	1110	IRQ14
0111	IRQ7	1111	IRQ15

3.3.19 DMH—DRAM MEMORY HOLE REGISTER

Address Offset: 68h
Default Value: 00h
Attribute: Read/Write
Size: 8 bits

The DMH Register defines a hole in main memory between 1 MByte and 16 MBytes. ISA memory accesses to the region defined by the memory hole are not forwarded to main memory. The ISA cycle is confined to the ISA Bus.

Bit	Description
7	**Memory Hole Enable (MHE):** When MHOLEE = 1, the memory hole is enabled and all ISA master and DMA accesses within the programmed hole are confined to the ISA Bus. All CPU and PCI master accesses within the hole are forwarded to the PCI/ISA Bus. When MHOLEE = 0 (default), the memory hole is disabled.
6:4	**Memory Hole Size (MHSIZE):** This field selects the memory hole size as shown in the table below.
3:0	**Memory Hole Start Address (MHSTRT):** This four bit field defines the starting address of the memory hole. Bits[3:0] correspond to A[23:20], respectively. The memory hole starting address can be between 1 MByte and 16 MBytes, with 1 MByte granularity. Note that the top of the memory hole range must be below 16 MBytes. It is the responsibility of the BIOS to set the hole size and starting address accordingly.

Bits[6:4]	Memory Hole Size	Bits[6:4]	Memory Hole Size
000	1 MByte (default)	100	Reserved
001	2 MBytes	101	Reserved
010	Reserved	110	Reserved
011	4 MBytes	111	8 MBytes

3.3.20 TOM—TOP OF MEMORY

Address Offset: 69h
Default Value: 02h
Attribute: Read/Write
Size: 8 bits

The 82420EX PCIset supports up to 128 MBytes of system memory. The Top Of Memory Register must be set by the BIOS to the value of the DRB4 Register plus the memory hole size. For example, the top of memory for a system with 16 MBytes of DRAMs, and a 1 MByte Hole (somewhere between 1 and 16 MBytes), is at 17 MBytes.

The TOM Register is programmed with an 8-bit upper address limit value. This upper address limit is compared to A[31:30,26:20] of the Host address bus to determine if main memory is being targeted. When A[31:30,26:20] < TOM, and the access is not to the memory hole, main memory is being targeted. Otherwise, a PCI or ISA region is being targeted. Bits[7:0] of this register correspond to A[31:30,26:20].

Note that SMRAM can be placed at the top of memory between TOM-64 KByte and TOM. Note, also, that the maximum supported DRAM size is 128 MBytes minus the Memory Hole size.

For use with operating systems other than Windows*, DOS*, and OS/2*, the TOM register should not be programmed with a value of greater than 127M.

*Other brands and names are the property of their respective owners.

3.3.21 SMRAMCON—SMRAM CONTROL REGISTER

Address Offset: 70h
Default Value: 00h
Attribute: Read/Write
Size: 8 Bits

The SMRAMCON Register sets the SMRAM location and attributes. SMRAM is always located in main memory and it is always non-cacheable. The memory block that shares the same bus address range with SMRAM is also non-cacheable (even if it is defined as cacheable by other configuration settings).

Bit	Description
7	**Reserved**
6	**SMRAM Space Open (SMOPN):** This bit manually opens SMRAM space. When SMOPN = 1, CPU accesses (only CPU accesses) to SMRAM are re-mapped (see SMBASE description). BIOS uses this bit to manually open the SMRAM when the CPU is not in SMM. SMOPN is used by the BIOS to initialize the SMRAM. Setting the SMOPN bit to 1 has no effect when the CPU is in SMM (SMLCK bit is 1) or when the CPU is not the current system master (HLDA = 1).
5	**SMRAM Close (SMCLS):** This bit manually closes SMRAM space. The SMI handler uses SMCLS to access the physical memory block that shares the same bus address range with SMRAM. When SMCLS = 1, re-mapping of SMRAM (code and data) is disabled. This permits the CPU to access the data in system memory that is aliased by SMM memory, even when the CPU is in SMM. Note that SMCLS affects data accesses only; code read cycles are not affected.
4	**SMRAM Lock (SMLCK):** SMLCK locks SMRAM space from manual opening. When SMLCK = 1, the SMOPN function is disabled, as well as write protecting the SMBASE. The SMLCK bit is a write once bit. This means that once set, this bit cannot be cleared by software. Only a CPURST clears this bit. SMLCK permits BIOS, after initialization is complete, to protect the SMRAM from other programs. Once SMLCK is set to 1, no manual opens of the SMRAM are possible.
3	**Reserved**
2:0	**SMRAM Base Address (SMBASE):** SMBASE selects the SMRAM segment. Based on this selection, the SMRAM address is re-mapped as shown below. The setting of the SMRAM range forces the CPU bus range to be non-cacheable, regardless of other bit settings. The following table describes some SMBASE values and attributes.

Bits[2:0]	CPU Range	DRAM Range	Non-CPU Cycles Are Sent to:	Comments
000				Reserved
001				Reserved
010	A0000–AFFFFh	A0000–AFFFFh	Subtractively to PCI/ISA	PCI/ISA graphic frame buffer region. Region can not be used as SMRAM if it is also used as a graphic frame buffer of a Host Bus device.
011	B0000–BFFFFh	B0000–BFFFFh		
100	C0000–CFFFFh	A0000–AFFFFh	Main memory or subtractively PCI/ISA, Function of registers setting.	
101	D0000–DFFFFh	A0000–AFFFFh		
110	E0000–EFFFFh	A0000–AFFFFh		
111	F0000–FFFFFh	A0000–AFFFFh		

3.3.22 SMICNTL—SMI CONTROL REGISTER

Address Offset: A0h
Default Value: 08h
Attribute: Read/Write
Size: 8 Bits

The SMICNTL Register provides Fast Off Timer control, STPCLK# enable/disable, and throttle control. This register also enables/disables the system management interrupt (SMI).

NOTE:
Bits[4:3] = 01 can be used to freeze the Fast Off Timer when in SMM. Freezing the Fast Off Timer prevents time-outs from occurring while executing SMM code. This prevents the system from being confused by asynchronous events that could happen while servicing SMM code.

Bit	Description
7:5	**Reserved**
4:3	**Fast Off Timer Control (CTMRCNTL):** This field enables/disables the Fast Off Timer and when enabled, selects the timer's counting granularity as shown below. **Bits[4:3]** / **Count Granularity for 33 MHz Host Bus Operation** / **Count Granularity for 25 MHz Host Bus Operation** 00 / 1 Minute / 1.32 Minutes 01 / Disabled (default) / Disabled (default) 10 / 1 HCLKIN / 1 HCLKIN 11 / 1 msec / 1.32 msec
2	**STPCLK# Signal Throttle Enable (CSTPCLKTHE):** This bit enables/disables control of the STPCLK# high/low times by the clock throttle timers. When bit 2 = 1, the STPCLK# signal throttle control is enabled. When enabled (and bit 1 = 1, enabling the STPCLK# signal), the high and low times for the STPCLK# signal are controlled by the Clock Throttle STPCLK# High Timer and Clock Throttle STPCLK# Low Timer Registers, respectively. When bit 2 = 0 (default), the throttle control of the STPCLK# signal is disabled.
1	**STPCLK# Signal Enable (CSTPCLKE):** This bit permits software to place the CPU into a low power state. When bit 1 = 1, the STPCLK# signal is enabled and a read from the APMC Register causes STPCLK# to be asserted. When bit 1 = 0 (default), the STPCLK# signal is disabled and is negated (high). Software can set this bit to 0 by writing a 0 to it.
0	**SMI# Gate (CSMIGATE):** When bit 0 = 1, the SMI# signal is enabled and a system management interrupt condition causes the SMI# signal to be asserted. When bit 0 = 0 (default), the SMI# signal is masked and negated. This bit only affects the SMI# signal and does not affect the detection/recording of SMI events (i.e., this bit does not affect the SMI status bits in the SMIREQ Register). Thus, SMI conditions can be pending when this bit is set to 1. If an SMI is pending when this bit is set to 1, the SMI# signal is asserted.

3.3.23 SMIEN—SMI ENABLE REGISTER

Address Offset: A2–A3h
Default Value: 0000h
Attribute: Read/Write
Size: 16 Bits

This register enables the generation of SMI (asserting the SMI# signal) for the associated hardware events (bits[5:0]), external SMI signal (bit 6), and software events (bit 7). When a hardware event is enabled, the occurrence of a corresponding event results in the assertion of SMI#, if enabled via the SMICNTL Register. The SMI# is asserted independent of the current power state (Power-On or Fast Off). The default for all sources in this register is disabled.

Bit	Description
15:8	**Reserved**
7	**APMC Write SMI Enable:** This bit enables SMI for writes to the APMC Register. When bit 7 = 1, writes to the APMC Register generate an SMI. When bit 7 = 0, writes to the APMC Register do not generate an SMI.
6	**EXTSMI# Signal SMI Enable:** When bit 6 = 1, asserting the EXTSMI# input signal generates an SMI. When bit 6 = 0, asserting EXTSMI# does not generate an SMI.
5	**Fast Off Timer SMI Enable:** This bit enables the Fast Off Timer to generate an SMI. When bit 5 = 1, the timer generates an SMI when it decrements to zero. When bit 5 = 0, the timer does not generate an SMI.
4	**IRQ12 SMI Enable (PS/2 Mouse Interrupt):** This bit enables the IRQ12 signal to generate an SMI. When bit 4 = 1, asserting the IRQ12 input signal generates an SMI. When bit 4 = 0, asserting IRQ12 does not generate an SMI.
3	**IRQ8 SMI Enable (RTC Alarm Interrupt):** This bit enables the IRQ8 signal to generate an SMI. When bit 3 = 1, asserting the IRQ8 input signal generates an SMI. When bit 3 = 0, asserting IRQ8 does not generate an SMI.
2	**IRQ4 SMI Enable (COM2/COM4 Interrupt or Mouse):** This bit enables the IRQ4 signal to generate an SMI. When bit 2 = 1, asserting the IRQ4 input signal generates an SMI. When bit 2 = 0, asserting IRQ4 does not generate an SMI.
1	**IRQ3 SMI Enable (COM1/COM3 Interrupt or Mouse):** This bit enables the IRQ3 signal to generate an SMI. When bit 1 = 1, asserting the IRQ3 input signal generates an SMI. When bit 1 = 0, asserting IRQ3 does not generate an SMI.
0	**IRQ1 SMI Enable (Keyboard Interrupt):** This bit enables the IRQ1 signal to generate an SMI. When bit 0 = 1, asserting the IRQ1 input signal generates an SMI. When bit 0 = 0, asserting IRQ1 does not generate an SMI.

3.3.24 SEE—SYSTEM EVENT ENABLE

Address Offset: A4–A7h
Default Value: 00000000h
Attribute: Read/Write
Size: 32 Bits

This register enables hardware events as system events or break events for power management control. Note that all of the functional bits in the SEE Register provide system event control. In addition, all bits provide break event control. The default for each system/break event in this register is disabled.

System Events: Activity by these events can keep the system from powering down. When a system event is enabled, the corresponding hardware event activity prevents a Fast Off powerdown condition. Anytime the corresponding hardware event occurs (signal is asserted), the Fast Off Timer is re-loaded with its initial count.

Break Events: These events can awaken a powered down system. When a break event is enabled, the corresponding hardware event activity powers up the system by negating STPCLK#. Note that STPCLK# is not negated until the stop grant special cycle has been generated by the CPU. Thus, from the time that STPCLK# is asserted until the stop grant cycle is returned, the occurrence of subsequent break events are latched in the IB.

NOTE:

Bit 30 in this register is used as a global break event and should be set to 1 if ISA cards that generate IRQ's by driving them low and then high and keeping them high until another interrupt is generated, are supported. Refer to the bit description.

SRESET is always enabled as a break event. However, SRESET only causes a break event after a stop grant special cycle has been received. If SRESET is asserted while STPCLK# is active and then negated before the stop grant cycle is received, SRESET does not cause a break event.

Bit	Description
31	**Fast Off SMI Enable (FSMIEN):** When bit 31 = 1 (enabled), an SMI causes a system event that re-loads the Fast Off Timer and a break event that negates the STPCLK# signal. When bit 31 = 0 (disabled), an SMI does not re-load the Fast Off Timer or negate the STPCLK# signal.
30	**INTR Enable (FINTREN):** When bit 30 of this register is set to 1, INTR will be used as a global break event. In this case, any IRQ that is generated will cause the system to power-up via the negation of STPCLK#, regardless of the state of bits 0, 1, and 3 through 15 in this register. When this bit is set to 0, INTR is not used as a break event and bits 0, 1, and 3 through 15 can be used to individually enable/disable break events. Note that this bit has no effect on the setting of system events and only effects the break event function.
29	**Fast Off NMI Enable (FNMIEN):** When bit 29 = 1 (enabled), an NMI (e.g., parity error) causes a system event that re-loads the Fast Off Timer and a break event that negates the STPCLK# signal. When bit 29 = 0 (disabled), an SMI does not re-load the Fast Off Timer or negate the STPCLK# signal.
28:16	**Reserved**

Bit	Description
15:3	**Fast Off IRQ[15:3] Enable:** These bits are used to prevent the system from entering Fast Off and break any current powerdown state when the selected hardware interrupt occurs. When a bit = 1 (enabled), the corresponding interrupt causes a system event that re-loads the Fast Off Timer and a break event that negates the STPCLK# signal. When a bit = 0 (disabled), the corresponding interrupt does not re-load the Fast Off Timer or negate the STPCLK# signal.
2	**Reserved**
1:0	**Fast Off IRQ[1:0] Enable:** These bits are used to prevent the system from entering Fast Off and break any current powerdown state when the selected hardware interrupt occurs. When a bit = 1, the corresponding interrupt causes a system event that re-loads the Fast Off Timer and a break event that negates the STPCLK# signal. When a bit = 0 (disabled), the corresponding interrupt does not re-load the Fast Off Timer or negate the STPCLK# signal.

3.3.25 FTMR—FAST OFF TIMER REGISTER

Address Offset: A8h
Default Value: 0Fh
Attribute: Read/Write
Size: 8 Bits

The Fast Off Timer is used to indicate (through an SMI) that the system has been idle for a pre-programmed period of time. The Fast Off Timer consists of a count-down timer and the count down value programmed into this register. The Fast Off Timer count down value is (x + 1) where x equals the value programmed in the Fast Off Timer register and the unit of measurement is in minutes or msec, depending on the value of bits 4–3 in the SMI Control register. The Fast Off Timer count down value is loaded into the Fast Off Timer when an enabled system event occurs. When the timer expires, an SMI special cycle is generated. When the Fast Off Timer is enabled (bits[4:3] = 00, 10, or 11 in the SMICNTL register), the timer counts down from the Fast Off Timer count down value. The count time interval is programmable (via the SMICNTL Register). When the Fast Off Timer reaches 00h, an SMI is generated and the timer is re-loaded with the Fast Off Timer count down value. If an enabled system event occurs before the Fast Off Timer reaches 00h, the Fast Off Timer is re-loaded with the Fast Off Timer count down value. Note that the Fast Off Timer should never be programmed to a value of 00h.

NOTE:
Before writing to the FTMR Register, the Fast Off Timer must be stopped by setting bits[4:3] to 01 in the SMICNTL Register. The Fast Off Timer will begin decrementing when the these bits are subsequently set to 00, 10, or 11.

Bit	Description
7:0	**Fast Off Timer Value:** Bits[7:0] contain value x, where the Fast Off Timer count down value is (x + 1). A read from the FTMR Register returns the value last written.

3.3.26 SMIREQ—SMI REQUEST REGISTER

Address Offset: AA–ABh
Default Value: 00h
Attribute: Read/Write
Size: 16 Bits

The SMIREQ Register contains status bits indicating the cause of an SMI. When an enabled event causes an SMI, the IB automatically sets the corresponding event's status bit to 1. Software sets the status bits to 0 by writing a 0 to them.

The SMI handler can query the status bits to see what caused the SMI and then branch to the appropriate routine. As the individual routines complete, the handler resets the appropriate status bit by writing a 0 to the corresponding bit.

Each of the SMIREQ bits is set by the IB in response to the activation of the corresponding SMI event. If the SMI event is still active when the corresponding SMIREQ bit is set to 0, the IB does not set the status bit back to a 1 (i.e., there is only one status indication per active SMI event).

When an IRQx signal is asserted, the corresponding RIRQx bit is set to a 1. If the IRQx signal is still active when software sets the RIRQx bit to 0, RIRQx is not set back to a 1. The IRQx may be negated before software sets the RIRQx bit to 0. If the RIRQx bit is set to 0 at the same time a new IRQx is activated, RIRQx remains set to 1. This indicates to the SMI handler that a new SMI event has been detected.

NOTE:
The SMIREQ bits are set, cleared, or read independently of each other and independently of the CSMIGATE bit in the SMICNTL Register.

Bit	Description
15:8	**Reserved**
7	**APM SMI Status (RAPMC):** The IB sets this bit to 1 to indicate that a write to the APM Control Register caused an SMI. Software sets this bit to a 0 by writing a 0 to it.
6	**EXTSMI# SMI Status (REXT):** The IB sets this bit to 1 to indicate that EXTSMI# caused an SMI. Software sets this bit to a 0 by writing a 0 to it.
5	**Fast Off Timer Expired Status (RFOT):** The IB sets this bit to 1 to indicate that the Fast Off Timer expired and caused an SMI. Software sets this bit to a 0 by writing a 0 to it.
4	**IRQ12 Request SMI Status (RIRQ12):** The IB sets this bit to 1 to indicate that IRQ12 caused an SMI. Software sets this bit to a 0 by writing a 0 to it.
3	**IRQ8# Request SMI Status:** The IB sets this bit to 1 to indicate that IRQ8# caused an SMI. Software sets this bit to a 0 by writing a 0 to it.
2	**IRQ4 Request SMI Status:** The IB sets this bit to 1 to indicate that IRQ4 caused an SMI. Software sets this bit to a 0 by writing a 0 to it.
1	**IRQ3 Request SMI Status:** The IB sets this bit to 1 to indicate that IRQ3 caused an SMI. Software sets this bit to a 0 by writing a 0 to it.
0	**IRQ1 Request SMI Status:** The IB sets this bit to 1 to indicate that IRQ1 caused an SMI. Software sets this bit to a 0 by writing a 0 to it.

3.3.27 CTLTMRL—CLOCK THROTTLE STPCLK# LOW TIMER

Address Offset: ACh
Default Value: 00h
Attribute: Read/Write
Size: 8 Bits

The duration of the STPCLK# asserted period when bit 2 in the SMICNTL Register is set to 1 is (x + 1) where x equals the value programmed in this register. The value in this register plus 1 is loaded into the STPCLK# Timer when STPCLK# is asserted. However, the timer does not start until the Stop Grant Bus Cycle is received. The STPCLK# timer counts using a 32 μs clock.

Bit	Description
7:0	**Clock Throttle STPCLK# Low Timer Value:** Bits[7:0] define the value x, where the Clock Throttle STPCLK# Low Timer count down value is (x + 1). (x + 1) defines the duration of the STPCLK# asserted period during clock throttling.

3.3.28 CTLTMRH—CLOCK THROTTLE STPCLK# HIGH TIMER

Address Offset: AEh
Default Value: 00h
Attribute: Read/Write
Size: 8 Bits

The duration of the STPCLK# negated period when bit 2 in the SMICNTL Register is set to 1 is (x + 1) where x equals the value programmed in this register. The value in this register plus 1 is loaded into the STPCLK# Timer when STPCLK# is negated. The STPCLK# timer counts using a 32 μs clock.

Bit	Description
7:0	**Clock Throttle STPCLK# High Timer Value:** Bits[7:0] define the value x, where the Clock Throttle STPCLK# High Timer count down value is (x + 1). (x + 1) defines the duration of the STPCLK# negated period during clock throttling.

3.4 ISA-Compatible Registers

This section describes the ISA-Compatible registers consisting of the DMA, interrupt controller, timer/counter, X-Bus control, NMI control, clock/reset, and advanced power management registers. Some of the registers are only accessible from the CPU/PCI Buses while others can be accessed by the CPU, PCI, or ISA Buses (see Table 7).

3.4.1 DMA REGISTER DESCRIPTION

The IB contains DMA circuitry that incorporates the functionality of two 82C37 DMA controllers. The two DMA controllers consist of two logical channel groupschannels [3:0] (Controller 1DMA1) and channels [7:4] (Controller 2DMA2).

This section describes the DMA registers. Unless otherwise stated, a CPURST sets each register to its default value. In addition, the DMA Master Clear Command (address 00Dh for channels [3:0] and 0DAh for channels [7:4]) permits software to set the DMA Command, DMA Status, DMA Request, and internal First/Last Flip-Flop Registers to their default values. The DMA Master Clear Command also sets the mask registers to their default values.

3.4.1.1 DCOM—DMA Command Register

I/O Address: Channels [3:0]—08h
Channels [7:4]—0D0h
Default Value: 00h
Attribute: Write Only
Size: 8 bits

This 8-bit register enables/disables the DMA channel groups, selects the priority scheme for responding to DMA requests, and selects the DMA request signal (DREQ) sense level. Following a CPURST or DMA Master Clear, both DMA1 and DMA2 are enabled in fixed priority and the DREQ sense level is active high.

Bit	Description
7	**DACK# Active Level (DACK[3:0,(7:5)]#):** Bit 7 controls the DMA channel request acknowledge (DACK#) assertion level. When bit 1 = 1, DACK# is an active high signal. When bit 1 = 0 (default), DACK# is an active low signal.
6	**DREQ Sense Assert Level (DREQ[3:0, (7:5)]):** Bit 6 controls the DREQx signal assertion level that the DMA controller detects as an active DMA channel request. Note that the DREQ channel assertion sensitivity is assigned by channel group, not per individual channel. When bit 6 = 0 (default), the DREQx sense assert level is active high. When bit 6 = 1, the DREQx sense assert level is active low. Following CPURST, the DREQx sense assert level is active high.
5	**Reserved:** Must be 0 when programming this register.
4	**DMA Group Arbitration Priority:** For priority resolution, the DMA consists of two logical channel groups—channels [3:0] (Controller 1—DMA1) and channels [7:4] (Controller 2—DMA2). Each group can be assigned fixed or rotating priority. Thus, both groups can be assigned fixed priority, one group can be assigned fixed priority and the other rotating priority, or both groups can be assigned rotating priority. When bit 4 = 0 (default), fixed priority is assigned to the channel. For fixed priority, the priority ordering is 0 (highest priority), 1, 2, 3, 5, 6, and 7 (lowest priority). Channels [3:0] of DMA-1 assume the priority position of Channel 4 in DMA-2, thus taking priority over channels 5, 6, and 7. Following CPURST, each group is initialized in fixed priority. When bit 4 = 1, rotating priority is assigned to the channel group. For rotating priority, the priority chain rotates so that the last channel serviced is assigned the lowest priority in the channel group ([3:0] or [7:5]). Channels [3:0] rotate as a group of 4. They are always placed between Channel 5 and Channel 7 in the priority list. Channel [7:5] rotate as part of a group of 4. That is, channels [7:5] form the first three positions in the rotation, while channels [3:0] comprise the fourth position in the arbitration.
3	**Reserved:** Must be 0 when programming this register.
2	**DMA Channel Group Enable:** When bit 2 = 1, the DMA channel group is disabled. Note that disabling channel group [7:4] also disables channel group [3:0], which is cascaded through channel 4. When bit 2 = 0 (default), the DMA channel group is enabled. Following CPURST, both channel groups are enabled.
1:0	**Reserved:** Must be 0 when programming this register.

3.4.1.2 DCM—DMA Channel Mode Register

I/O Address: Channels [3:0]—0Bh
Channels [7:4]—0D6h
Default Value: Bits[7:2] = 0, Bits[1:0] = undefined
Attribute: Write Only
Size: 8 bits

The Channel Mode Register controls DMA transfer type, transfer mode, address increment/decrement, and autoinitialization. The DMA transfer mode for channel 4 defaults to cascade and cannot be programmed for any mode other than DMA transfer mode.

Bit	Description
7:6	**DMA Transfer Mode:** Bits[7:6] select the DMA transfer mode as shown below. **Bits[7:6]** — **DMA Transfer Mode** 00 — Demand (default) 01 — Single 10 — Block 11 — Cascade
5	**Address Increment/Decrement Select:** Bit 5 controls address increment/decrement during multi-byte DMA transfers. When bit 5 = 0 (default), address increment is selected. When bit 5 = 1, address decrement is selected.
4	**Autoinitialize Enable:** When bit 4 = 1, the DMA restores the base page, address, and word count information to their respective current registers following a terminal count (TC). When bit 4 = 0 (default), the autoinitialize feature is disabled and the DMA does not restore the above mentioned registers.
3:2	**DMA Transfer Type:** This field selects verify, write, or read data transfer types as shown below. Write transfers move data from an I/O device to memory. Read transfers move data from memory to an I/O device. Verify transfers are pseudo transfers; addresses are generated as in a normal read or write transfer. However, with Verify transfers, the ISA memory and I/O cycle lines are not driven. When the channel is programmed for cascade (bits[7:6] = 11), the transfer type bits are irrelevant. **Bits[3:2]** — **DMA Transfer Type** 00 — Verify (default) 01 — Write 10 — Read 11 — Illegal: do not write
1:0	**DMA Channel Select:** This field select the DMA Channel Mode Register that will be written by bits[7:2] as shown below. These bits are undefined after a hard reset. **Bits[1:0]** — **DMA Channel Select** 00 — Channel 0(4) 01 — Channel 1(5) 10 — Channel 2(6) 11 — Channel 3(7)

3.4.1.3 DREQ—DMA Request Register

I/O Address: Channels [3:0]—09h
Channels [7:4]—0D2h
Default Value: Bits[1:0] = undefined, Bits[7:2] = 0
Attribute: Write Only
Size: 8 bits

The DMA Request Register is used by software to initiate a DMA request. The DMA responds to the software request as though DREQx is asserted. These requests are non-maskable and subject to prioritization by the priority encoder. When a TC is generated, the channel's request bit is set to 0. For software DMA requests, the channel must be in Block Mode. Note that the DMA Request Register status for DMA1 and DMA2 can be obtained from bits[7:4] of the DMA Status Register. The request bit for each channel is set to its default value by a CPURST or a Master Clear. The register is not affected by the RSTDRV output.

Bit	Description
7:3	**Reserved:** Must Be 0 when programming this register.
2	**DMA Channel Service Request:** When bit 2 = 1, a software DMA transfer is requested for the channel specified by bits[1:0]. When bit 2 = 0 (default), software DMA transfers are not requested for the channel specified by bits[1:0].
1:0	**DMA Channel Select:** This field selects the DMA channel to be written by bit 2 as shown below. **Bits[1:0] — DMA Channel Select** 00 — Channel 0(4) 01 — Channel 1(5) 10 — Channel 2(6) 11 — Channel 3(7)

3.4.1.4 WSMB—Write Single Mask Bit Register

I/O Address: Channels [3:0]—0Ah
Channels [7:4]—0D4h
Default Value: Bits[1:0] = undefined, Bit 2 = 1, Bits[7:3] = 0
Attribute: Write Only
Size: 8 bits

The WSMB Register permits the masking of the incoming DMA requests (DREQx) for each channel. A channel's mask bit is automatically set when the Current Byte/Word Count Register reaches terminal count, unless the channel is programmed for autoinitialization. This register is set to its default value by a CPURST or a Master Clear. Setting the entire register disables all DMA requests until a Clear Mask Register instruction allows them to occur. This instruction format is similar to the format used with the DMA Request Register.

NOTE:
Individually masking DMA channel 4 (DMA controller 2, channel 0) automatically masks DMA channels [3:0], as this channel group is logically cascaded onto channel 4. Setting this mask bit disables the incoming DREQ's for channels [3:0].

Bit	Description
7:3	**Reserved:** Must be 0 when programming this register.
2	**Channel Mask Select:** When bit 2 = 1 (default), DREQ is masked (disabled) for the channel selected by bits[1:0]. When bit 2 = 0, DREQ is not masked (enabled) for the channel selected by bits[1:0].
1:0	**DMA Channel Select:** This field selects the DMA channel to be written by bit 2 as shown below. **Bits[1:0] — DMA Channel Select** 00 — Channel 0(4) 01 — Channel 1(5) 10 — Channel 2(6) 11 — Channel 3(7)

3.4.1.5 WAMB—Write All Mask Bits Register

I/O Address: Channels [3:0]—0Fh
Channels [7:4]—0DEh
Default Value: Bit[3:0] = 1, Bit[7:4] = 0
Attribute: Read/Write
Size: 8 bits

This register enables/disables the incoming DREQx signals. All four channels can be simultaneously enabled/disabled instead of enabling/disabling each channel individually, as is the case with the Write Single Mask Bit Register.

Unlike the WSMB Register, the WAMB Register includes a status read to check the current mask status of the selected DMA channel group. A channel's mask bit is automatically set to 1 when the Current Byte/Word Count Register reaches terminal count, unless the channel is programmed for autoinitialization. All mask bits are set to 1 (disable) by CPURST or a Master Clear. Setting these bits to 1 disables all DMA requests until a Clear Mask Register instruction enables the requests.

NOTES:

1. Individually masking DMA channel 4 (DMA controller 2, channel 0) automatically masks DMA channels [3:0], as this channel group is logically cascaded onto channel 4.
2. Masking DMA controller 2 with a write to address 0DEh also masks DREQ assertions from the DMA controller, as this channel group is logically cascaded onto channel 4. When DMA channel 4 is masked, so are DMA channels [3:0].

Bit	Description
7:4	**Reserved:** Must be 0 when programming this register.
3:0	**Channel Mask Bits:** Setting the bit(s) to a 1 (default) disables the corresponding DREQ(s). Setting the bit(s) to a 0 enables the corresponding DREQ(s). When read, bits[3:0] indicate the DMA channel [3:0] ([7:4]) mask status.

3.4.1.6 DS—DMA Status Register

I/O Address: Channels [3:0]—08h
Channels [7:4]—0D0h
Default Value: 00h
Attribute: Read Only
Size: 8 bits

This register indicates which channels have reached terminal count and which channels have a pending DMA request.

Bit	Description
7:4	**Channel Request Status:** When a valid DMA request is pending for a channel (on its DREQ signal line), the corresponding bit is set to 1. When a DMA request is not pending for a particular channel, the corresponding bit is set to 0. The source of the DREQ may be hardware or a software request. Note that channel 4 does not have DREQ or DACK lines, so the response for a read of DMA2 status for channel 4 is irrelevant.
3:0	**Channel Terminal Count Status:** When a channel reaches terminal count (TC), its status bit is set to 1. If TC has not been reached, the status bit is set to 0. Note that channel 4 is programmed for cascade, and is not used for a DMA transfer. Therefore, the TC bit response for a status read on DMA2 for channel 4 is irrelevant. Bits[3:0] are set to 0 upon CPURST and on a read of the DS Register.

3.4.1.7 DB&CA—DMA Base And Current Address Registers (8237 Compatible Segment)

I/O Address: DMA Channel 0—000h, DMA Channel 1—002h, DMA Channel 2—004h,
DMA Channel 3—006h, DMA Channel 4—0C0h, DMA Channel 5—0C4h,
DMA Channel 6—0C8h, DMA Channel 7—0CCh
Default Value: Undefined
Attribute: Read/Write
Size: 16 bits per channel

Each channel has a 16-bit Current Address Register. This register contains the value of the 16 least significant bits of the full 27-bit address used during DMA transfers. The address is automatically incremented or decremented after each transfer and the intermediate values of the address are stored in the Current Address Register during the transfer. The Host CPU reads/writes the register in successive 8-bit bytes. This register is not accessible by ISA Bus masters. The programmer must issue the Clear Byte Pointer Flip-Flop Command to reset the internal byte pointer and correctly align the write prior to programming the Current Address Register. After clearing the Byte Pointer Flip-Flop, the first read/write accesses the low byte (bits[7:0]), and the second read/write accesses the high byte (bits[15:8]). Note that a mixed sequence of read and write cycles continues to toggle the Byte Pointer Flip-Flop, and successive reads and writes from this register alternate between the low byte and the high byte. An autoinitialize re-initializes the Current Address Register back to its original value following a TC. Autoinitialize occurs only after a TC.

Each channel has a Base Address Register located at the same address as the corresponding Current Address Register. These registers store the original value of their associated Current Address Registers. During autoinitialize these values are used to restore the Current Address Registers to the original values. The Base Registers are written simultaneously with their corresponding Current Address Register in successive 8-bit bytes. The Base Registers are write only.

Bit	Description
15:0	**Base and Current Address[15:0]:** These bits represent the 16 least significant address bits used during DMA transfers. Together with the DMA Low Page Register, they form the ISA-Compatible 24-bit DMA address. As an extension of the ISA-Compatible functionality, the DMA High Page Register completes the 27-bit DMA address generation, supporting DMA transfers throughout the full 128 MBytes of main memory. Upon CPURST or Master Clear, the value of these bits are undefined.

3.4.1.8 DB&CBW—DMA Base And Current Byte/Word Count Registers (8237 Compatible Segment)

I/O Address: DMA Channel 0—001h, DMA Channel 1—003h, DMA Channel 2—005h, DMA Channel 3—007h, DMA Channel 4—0C2h, DMA Channel 5—0C6h, DMA Channel 6—0CAh, DMA Channel 7—0CEh
Default Value: Undefined
Attribute: Read/Write
Size: 16 bits per channel

Each channel has a 16-bit Current Byte/Word Count Register that determines the number of transfers to be performed. The actual number of transfers is one more than the number programmed in the Current Byte/Word Count Register (i.e., programming a count of 100 results in 101 transfers). The byte/word count is decremented after each transfer. The intermediate value of the byte/word count is stored in the register during the transfer. When the value in the register goes from 0000h to FFFFh, a TC is generated.

Following the end of a DMA service, the register may also be re-initialized by an autoinitialization back to its original value. Autoinitialize can only occur when a TC occurs. If it is not autoinitialized, this register has a count of FFFFh after TC.

For transfers to/from an 8-bit I/O, the byte/word count indicates the number of bytes to be transferred. This applies to DMA channels [3:0]. For transfers to/from a 16-bit I/O, with shifted address, the byte/word count indicates the number of 16-bit words to be transferred. This applies to DMA channels [7:5].

Each channel has a Base Byte/Word Count Register located at the same I/O address as the corresponding Current Byte/Word Count Register. These registers store the original value of their associated Current Byte/Word Count Registers. During autoinitialize, these values are used to restore the Current Registers to their original values. The Base Registers are written simultaneously with their corresponding Current Register in successive 8-bit bytes. The Base Registers cannot be read by external agents.

Bit	Description
15:0	**Base and Current Byte/ Word Count:** These bits represent the 16 byte/word count bits used when counting down a DMA transfer. Upon CPURST or Master Clear, the value of these bits are undefined.

3.4.1.9 DMLPG—DMA Memory Low Page Registers

I/O Address: DMA Channel 0—087h, DMA Channel 1—083h, DMA Channel 2—081h, DMA Channel 3—082h, DMA Channel 5—08Bh, DMA Channel 6—089h, DMA Channel 7—08Ah
Default Value: Undefined
Attribute Read/Write
Size: 8 bits per channel

Each channel has an 8-bit Low Page Register. The DMA memory Low Page Register contains bits[23:16] of the 27-bit address. The register works in conjunction with the DMA controller's High Page Register and Current Address Register to define the complete address (27 bits) for the DMA channel. This register is static throughout the DMA transfer. Following an autoinitialization, this register retains the original programmed value. Autoinitialize takes place only after a TC.

Bit	Description
7:0	**DMA Low Page Address Bits[23:16]:** These bits represent the eight second most significant address bits when forming the 27-bit address for a DMA transfer. Upon CPURST or Master Clear, the value of these bits are undefined.

3.4.1.10 DMHPG—DMA Memory High Page Register

I/O Address: DMA Channel 0—487h, DMA Channel 1—483h, DMA Channel 2—481h, DMA Channel 3—482h, DMA Channel 5—48Bh, DMA Channel 6—489h, DMA Channel 7—48Ah
Default Value: Undefined
Attribute Read/Write
Size: 8 bits per channel

Each channel has an 8-bit High Page Register. The DMA Memory High Page Register contains the three most significant bits of the 27-bit address. The register works in conjunction with the Current Address Register and Low Page Register to define the complete 27-bit address for the DMA channel. This register is static throughout the DMA transfer. Following an autoinitialization, this register retains the original programmed value. Autoinitialize occurs only after a TC.

Bit	Description
7:3	**Reserved:** Must be 0 when programming this register.
2:0	**DMA High Page [26:24]:** These bits represent the three most significant address bits when forming the 27-bit address for a DMA transfer. Following the programming of a channel's Current Address Register or Low Page Register, this register is initialized to 00h. Upon CPURST or Master Clear, the value of these bits are undefined.

3.4.1.11 DCLBP—DMA Clear Byte Pointer Register

I/O Address: Channels [3:0]—00Ch
Channels [7:4]—0D8h
Default Value: Undefined
Attribute: Write Only
Size: 8 bits

Writing to this register executes the Clear Byte Pointer Command. This command is executed prior to writing/reading new address or word count information to/from the DMA. This command initializes the byte pointer flip-flop to a known state so that subsequent byte accesses to the 16-bit register contents address upper and lower bytes in the correct sequence.

The clear byte pointer command clears the internal flip-flop used to address the upper or lower byte of the 16-bit Address and Word Count Registers. The latch is also cleared at power-on by CPURST and by the Master Clear Command. The Host CPU may read or write a 16-bit DMA controller register by performing two consecutive accesses to the I/O port. The Clear Byte Pointer Command precedes the first access. The first I/O write to the register address loads the least significant byte, and the second access automatically accesses the most significant byte.

Bit	Description
7:0	**Clear Byte Pointer:** No specific pattern. The command is invoked with a write to the I/O address.

3.4.1.12 DMCL—DMA Master Clear Register

I/O Address: Channel [3:0]—00Dh
Channel [7:4]—0DAh
Default Value: Undefined
Attribute: Write Only
Size: 8 bit

This software command has the same effect as the hardware reset. The Command, Status, Request, and Internal First/Last Flip-Flop registers are cleared and the Mask Register is set. The DMA controller enters the idle cycle.

Bit	Description
7:0	**Master Clear Command:** No specific pattern. This command is invoked with a write to the I/O address.

3.4.1.13 DCLM—DMA Clear Mask Register

I/O Address: Channel [3:0]—00Eh
Channel [7:4]—0DCh
Default Value: Undefined
Attribute: Write Only
Size: 8 bit

This command clears the mask bits of all four channels, enabling them to accept DMA requests. I/O port 0Eh is used for channels 0–3 and I/O port 0DCh is used for channels 4–7.

Bit	Description
7:0	**Clear Mask Register Command:** No specific pattern. This command is invoked with a write to the I/O port address.

3.4.2 TIMER/COUNTER REGISTER DESCRIPTION

There are three counters that are equivalent to those found in the 82C54 Programmable Interval Timer. The counters are controlled by timer/counter registers that can be accessed from either the CPU, PCI Bus, or ISA Bus.

3.4.2.1 TCW—Timer Control Word Register

I/O Address: 043h
Default Value: Undefined
Attribute: Write Only
Size: 8 bits

The Timer Control Word Register specifies the counter selection, the operating mode, the counter byte programming order and size of the count value, and whether the counter counts down in a 16-bit binary or binary-coded decimal (BCD) format. After writing the control word, a new count can be written at any time. The new value takes effect according to the programmed mode.

There are six programmable counting modes. Typically, Timer Counters 0 and 2 are programmed for Mode 3, the Square Wave Mode, while Counter 1 is programmed in Mode 2, the Rate Generator Mode.

Two latch commands are selected through the Timer Control Word Register. The Read Back Command is selected when bits[7:6] = 11 and the Counter Latch Command is selected when bits[5:4] = 00. When either of these two commands are selected, the meaning of the other bits in the register changes.

Following CPURST, the control words for each register are undefined and each timer must be programmed for the counters to be in a known state. Note however, that some counter/timer functions are set to known states following CPURST. Each counter OUT signal is set to 0 (and the Timer Counter 2 OUT status bit in the NMISC Register is 0). The SPKR output, interrupt controller input IRQ0 (internal), and the internally generated refresh request are each set to 0 following CPURST.

Bit	Description
7:6	**Counter Select:** The Counter Selection bits select the counter the control word acts upon or the Read Back Command as shown below. **Bits[7:6]** — **Counter Select** 00 — Counter 0 01 — Counter 1 10 — Counter 2 11 — Read Back Command
5:4	**Read/Write Select:** This field selects the count register read/write programming mode or the Counter Latch Command as shown below. The read/write programming selection chosen indicates the programming sequence that must follow when initializing the counter specified in bits[7:6]. If a counter is programmed to read/write two byte counts, note that a program must not transfer control between writing the first and second byte to another routine that also writes into that same counter. Otherwise, the counter will be loaded with an incorrect count. The count must be completely loaded with both bytes. Note that the actual counter programming occurs by accessing I/O addresses 040h, 041h, and 042h for counters 0, 1, and 2, respectively. **Bits[5:4]** — **Read/Write Programming Select** 00 — Counter Latch Command 01 — R/W Least Significant Byte (LSB) 10 — R/W Most Significant Byte (MSB) 11 — R/W LSB, Then MSB
3:1	**Counter Mode Selection:** This field selects one of six possible modes of operation for the counter as shown below. **Bits[3:1]** — **Counter Mode** 000 — Out Signal On End of Count 001 — Hardware Re-triggerable one-shot X10 — Rate Generator (divide by n counter) X11 — Square Wave Output 100 — Software Triggered Strobe 101 — Hardware Triggered Strobe
0	**Binary/BCD Countdown Select:** When bit 0 = 0, a binary countdown is used. The largest possible binary count is 2^{16}. When bit 0 = 1, a binary-coded decimal (BCD) count is used. The largest BCD count allowed is 10^4.

Read Back Command

The Read Back Command provides the count value, programmed mode, and current states of the OUT pin and Null count flag of the selected counter or counters. The Read Back Command is written to the Timer Control Word Register that latches the current states of the above mentioned variables. The value of the counter and its status may then be read by I/O access to the counter address.

Status and/or count may be latched on one, two, or all three of the counters by selecting the counter during the register write. The count latched remains latched until read, regardless of further latch commands. The count must be read before newer latch commands latch a new count. The status latched by the Read Back Command also remains latched until after a read of the Counter Access Ports Register. Thus, the status and count are unlatched only after a counter read of the Timer Status Byte Format Register, the Counter Access Ports Register, or the Timer Status Byte Register and Counter Access Ports Register in succession.

Both count and status of the selected counter(s) may be latched simultaneously by setting both bit 5 and bit 4 to 0. This is functionally the same as issuing two consecutive, separate Read Back Commands. As mentioned above, if multiple count and/or status Read Back Commands are issued to the same counter(s) without any intervening reads, all but the first are ignored.

If both count and status of a counter are latched, the first read operation from that counter returns the latched status, regardless of which was latched first. The next one or two reads (depending on whether the counter is programmed for one or two byte counts) returns the latched count. Subsequent reads return an unlatched count.

Bit	Description
7:6	**Read Back Command:** When bits[7:6] = 11 during a write to the Timer Control Word Register, the Read Back Command is selected. As noted above, the normal meanings (mode, countdown, R/W select) of the bits in the control register at I/O address 043h change when the Read Back Command is selected. Following the Read Back Command, I/O reads from the selected counter's I/O addresses produce the current latch status, the current latched count, or both if bits[5:4] = 00.
5	**Latch Count of Selected Counters:** When bit 5 = 1, the count is not latched. When bit 5 = 0, the current count value of the selected counters is latched.
4	**Latch Status of Selected Counters:** When bit 4 = 1, the status is not latched. When bit 4 = 0, the status of the selected counters is latched. The status byte format is described in Section 3.4.2.2, Interval Timer Status Byte Format Register.
3	**Counter 2 Select:** When bit 3 = 1, Counter 2 is selected for the latch command selected with bits 4 and 5. When bit 3 = 0, status and/or count is not latched.
2	**Counter 1 Select:** When bit 2 = 1, Counter 1 is selected for the latch command selected with bits 4 and 5. When bit 2 = 0, status and/or count is not latched.
1	**Counter 0 Select:** When bit 1 = 1, Counter 0 is selected for the latch command selected with bits 4 and 5. When bit 1 = 0, status and/or count is not latched.
0	**Reserved:** Must be 0 when programming this register.

Counter Latch Command

The Counter Latch Command latches the current count value at the time the command is issued. This command is used to insure that the count read from the counter is accurate (particularly when reading a two-byte count). The count value is then read from each counter's count register (via the Counter Access Ports Register). One, two, or all three counters may be latched with one Counter Latch Command.

If a Counter is latched once and then, some time later, latched again before the count is read, the second Counter Latch Command is ignored. The count read is the count at the time the first Counter Latch Command was issued.

The count must be read according to the programmed format. Specifically, if the counter is programmed for two byte counts, two bytes must be read. The two bytes do not have to be read one right after the other (read, write, or programming operations for other counters may be inserted between the reads).

NOTES:

1. If a counter is programmed to read/write two byte counts, a program must not transfer control between reading the first and second byte to another routine that also reads from that same counter. Otherwise, an incorrect count will be read. Finish reading the latched two-byte count before transferring control to another routine.
2. The Timer Counter Register bit definitions are different during the Counter Latch Command than for a normal Timer Counter Register write.

Bit	Description
7:6	**Counter Selection:** This field selects the counter for latching by the Counter Latch Command as shown below. **Bits[7:6]** — **Counter Select** 00 — Counter 0 01 — Counter 1 10 — Counter 2 11 — Not Used; Do Not Write
5:4	**Counter Latch Command:** When bits[5:4] = 00 during a write to the Timer Control Word Register, the Counter Latch Command is selected. As noted above, the normal meanings (mode, countdown, R/W select) of the bits in the control register at I/O address 043h change when the Counter Latch Command is selected. Following the Counter Latch Command, I/O reads from the selected counter's I/O addresses produce the current latched count.
3:0	**Reserved:** Must be 0 when programming this register.

3.4.2.2 TMSTAT—Interval Timer Status Byte Format Register

I/O Address: Counter 0—040h
Counter 1—041h
Counter 2—042h
Default Value: Bits[6:0] = undefined, Bit 7 = 0
Attribute: Read Only
Size: 8 bits per counter

Each counter's status byte can be read following an Interval Timer Read Back Command. The Read Back Command is programmed through the Timer Control Word Register. If latch status is chosen (bit 4 = 0, Read Back Command) as a read back option for a given counter, the next read from the counter's Counter Access Ports Register returns the status byte. The status byte returns the countdown type (either BCD or binary), the counter operational mode, the read/write selection status, the Null count (also referred to as the count register status), and the current state of the counter OUT pin.

Bit	Description
7	**Counter OUT Pin State:** When bit 7 = 1, the OUT pin of the counter is 1. When bit 7 = 0 (default) , the OUT pin of the counter is 0.
6	**Count Register Status:** Null Count (also referred to as the Count Status Register) indicates when the last count written to the Count Register (CR) has been loaded into the counting element (CE). The exact time this happens depends on the counter mode. However, until the count is loaded into the counting element (CE), it can't be read from the counter. If the count is latched or read before the load time, the count value returned does not reflect the new count written to the register. When bit 6 = 0, the count has been transferred from CR to CE and is available for reading. When bit 6 = 1, the Null count condition exists. The count has not been transferred from CR to CE and is not yet available for reading.
5:4	**Read/Write Selection Status:** Bits[5:4] reflect the read/write selection made through bits[5:4] of the Timer Control Word Register. The binary codes returned during the status read match the codes used to program the counter read/write selection. **Bits[5:4] — Read/Write Programming Select Status** 00 — Counter Latch Command 01 — R/W Least Significant Byte (LSB) 10 — R/W Most Significant Byte (MSB) 11 — R/W LSB, Then MSB
3:1	**Mode Selection Status:** Bits[3:1] return the counter mode programming made through bits[3:1] of the Timer Control Word Register. The binary code returned matches the code used to program the counter mode, as listed under the bit function above. **Bits[3:1] — Counter Mode** 000 — Out Signal On End of Count 001 — Hardware Re-triggerable one-shot X10 — Rate Generator (divide by n counter) X11 — Square Wave Output 100 — Software Triggered Strobe 101 — Hardware Triggered Strobe
0	**Countdown Type Status:** Bit 0 reflects the current countdown type—either 0 for binary countdown or a 1 for binary coded decimal (BCD) countdown.

3.4.2.3 CAPS—Counter Access Ports Register

I/O Address: Counter 0, System Timer—040h
Counter 1, Refresh Request—041h
Counter 2, Speaker Tone—042h
Default Value: Undefined
Attribute: Read/Write
Size: 8 bits per counter

These I/O addresses provide access for a) writing count values to the Count Registers, b) reading the current count value from the counter by either an I/O read, after a counter-latch command, or after a Read Back Command, and c) reading the status byte following a Read Back Command.

Bit	Description
7:0	**Counter Port Byte:** Each counter I/O address is used to program the 16-bit Count Register. The order of programming, either LSB only, MSB only, or LSB then MSB, is defined by the Timer Control Word Register. This register I/O address also reads the current count from the Count Register and returns the status of the counter programming following a Read Back Command.

3.4.3 INTERRUPT CONTROLLER REGISTER DESCRIPTION

The 82420EX PCIset contains an ISA-Compatible interrupt controller that incorporates the functionality of two 82C59 interrupt controllers. The interrupt registers control the operation of the interrupt controller and can be accessed from the CPU or PCI Bus. In addition, some of the registers can be accessed from the ISA Bus.

3.4.3.1 ICW1—Initialization Command Word 1 Register

I/O Address: INT CNTRL-1—020h
INT CNTRL-2—0A0h
Default Value: Undefined
Attribute: Write Only
Size: 8 bits per controller

A write to Initialization Command Word 1 starts the interrupt controller initialization sequence. Addresses 020h and 0A0h are referred to as the base addresses of CNTRL-1 and CNTRL-2, respectively. An I/O write to the CNTRL-1 or CNTRL-2 base address with bit 4 equal to 1 is interpreted as ICW1. For the 82420EX PCIset-based ISA systems, three I/O writes to ''base address + 1'' must follow the ICW1. The first write to ''base address + 1'' performs ICW2, the second write performs ICW3, and the third write performs ICW4. ICW1 starts the initialization sequence during which the following automatically occur:

1. The Interrupt Mask Register is cleared.
2. IRQ7 input is assigned priority 7.
3. The slave mode address is set to 7.
4. Special Mask Mode is cleared and Status Read is set to IRR.
5. If IC4 was set to 0, then all functions selected by ICW4 are set to 0. However, in the IB implementation, ICW4 must be programmed and IC4 must be set to a 1.

ICW1 has three significant functions in the IB interrupt controller configuration. ICW4 is needed, so bit 0 must be programmed to a 1. Since there are two interrupt controllers in the system, bit 1 (SNGL) must be programmed to a 0 on both CNTRL-1 and CNTRL-2, to indicate a cascade configuration. The IB provides separate registers (ELCR Registers) to program level or edge sensitive interrupt IRQ lines. Thus, bit 3 (LTIM in the 82C59) is not used. Bit 4 must be a 1 when programming ICW1. This bit indicates that ICW1, and not OCW2 or OCW3, will be programmed during the write to this port.

Bit	Description
7:5	**ICW/OCW Select:** Bits[7:5] are MCS-85 implementation specific bits. These bits are not used and should be 000 when programming the interrupt controller.
4	**ICW/OCW Select:** Bit 4 = 1 selects ICW1. OCW2 and OCW3 are also addressed at the same port as ICW1. A ''1'' on this bit at any time will force the interrupt controller to interpret the write as an ICW1. The controller will then expect to see ICW2, ICW3, and ICW4. After the fixed initialization sequence to ICW1, ICW2, ICW3, and ICW4, the controller base address is used to write to OCW2 and OCW3. Bit 4 is a ''0'' on writes to these registers.
3	**Edge/Level Bank Select (LTIM):** This bit is disabled. Its 82C59 Interrupt Controller function is replaced by the Edge/Level Triggered Control Registers (ELCR). The ELCR registers allow the interrupts to be programmed for edge or level mode on an interrupt by interrupt basis.
2	**ADI:** Bit 2 (ADI) is a MCS-85 implementation specific bit. This bit is not used and should be 0 when programming the IB.
1	**Single/Cascade Select (SNGL):** SNGL must be programmed to a 0 to indicate that two interrupt controllers are operating in cascade mode on the IB.
0	**ICW4 Write Required (ICW4):** This bit must be set to a 1. IC4 indicates that ICW4 needs to be programmed. The IB requires that ICW4 be programmed to indicate that the controllers are operating in an 80x86 type system.

3.4.3.2 ICW2—Initialization Command Word 2 Register

I/O Address: INT CNTRL-1—021h
INT CNTRL-2—0A1h
Default Value: Undefined
Attribute: Write Only
Size: 8 bits per controller

ICW2 is used to initialize the interrupt controller with the five most significant bits of the interrupt vector address. The value programmed for bits[7:3] is used by the Host CPU to define the base address in the interrupt vector table for the interrupt routines associated with each IRQ on the controller. Typical ISA ICW2 values are 08h for CNTRL-1 and 70h for CNTRL-2.

Bit	Description
7:3	**Interrupt Vector Base Address:** Bits[7:3] define the base address in the interrupt vector table for the interrupt routines associated with each interrupt request level input. For CNTRL-1, a typical value is 00001b, and for CNTRL-2, 01110b. The interrupt controller combines a binary code representing the interrupt level to receive service with this base address to form the interrupt vector that is driven onto the bus. For example, the complete interrupt vector for IRQ0 (CNTRL-1), would be 0000 1000b (CNTRL-1 [7:3] = 00001b and 000b representing IRQ0). This vector is used by the CPU to point to the address information that defines the start of the interrupt routine.
2:0	**Interrupt Request Level:** When writing ICW2, this field should be 000. During an interrupt acknowledge cycle, this field is programmed by the interrupt controller with the interrupt code representing the interrupt level to be serviced. This interrupt code is combined with bits[7:3] to form the complete interrupt vector driven onto the data bus during the second INTA# cycle. The 3-bit binary codes are: 000 represents IRQ0 (IRQ8), 001 IRQ1 (IRQ9), 010 IRQ2 (IRQ10), 011 IRQ3 (IRQ11), 100 IRQ4 (IRQ12), 101 IRQ5(IRQ13), 110 IRQ6(IRQ14), and 111 IRQ7 (IRQ15).

3.4.3.3 ICW3—Initialization Command Word 3 Register

I/O Address: INT CNTRL-1—021h
Default Value: Undefined
Attribute: Write Only
Size: 8 bits

The meaning of ICW3 differs between CNTRL-1 and CNTRL-2. On CNTRL-1, the master controller, ICW3 indicates which CNTRL-1 IRQ line physically connects the INTR output of CNTRL-2 to CNTRL-1. ICW3 must be programmed to 04h, indicating the cascade of the CNTRL-2 INTR output to the IRQ2 input of CNTRL-1.

An interrupt request on IRQ2 causes CNTRL-1 to enable CNTRL-2 to present the interrupt vector address during the second interrupt acknowledge cycle.

Bit	Description
7:3	**Not Used:** Must be 0.
2	**Cascaded Interrupt Controller IRQ Connection:** Bit 2 must always be programmed to a 1. This bit indicates that CNTRL-2, the slave controller, is cascaded on interrupt request line two (IRQ2). When an interrupt request is asserted to CNTRL-2, the IRQ goes through the priority resolver. After the slave controller priority resolution is finished, the INTR output of CNTRL-2 is asserted. However, this INTR assertion does not go directly to the CPU. Instead, the INTR assertion cascades into IRQ2 on CNTRL-1. IRQ2 must go through the priority resolution process on CNTRL-1. If it wins the priority resolution on CNTRL-1 and the CNTRL-1 INTR signal is asserted to the CPU, the returning interrupt acknowledge cycle is really destined for CNTRL-2. The interrupt was originally requested at CNTRL-2, so the interrupt acknowledge is destined for CNTRL-2, and not a response for IRQ2 on CNTRL-1. When an interrupt request from IRQ2 wins the priority arbitration, in reality an interrupt from CNTRL-2 has won the arbitration. Because bit 2 of ICW3 on the master is set to 1, the master knows which identification code to broadcast on the internal cascade lines, alerting the slave controller that it is responsible for driving the interrupt vector during the second INTA# pulse.
1:0	**Not Used:** Must be 0.

3.4.3.4 ICW3—Initialization Command Word 3 Register

I/O Address: INT CNTRL-2—0A1h
Default Value: Undefined
Attribute: Write Only
Size: 8 bits

On CNTRL-2 (the slave controller), ICW3 is the slave identification code broadcast by CNTRL-1 from the trailing edge of the first INTA# pulse to the trailing edge of the second INTA# pulse. CNTRL-2 compares the value programmed in ICW3 with the incoming identification code. The code is broadcast over three internal cascade lines. ICW3 must be programmed to 02h for CNTRL-2. When 010b is broadcast by CNTRL-1 during the INTA# sequence, CNTRL-2 assumes responsibility for broadcasting the interrupt vector during the second interrupt acknowledge cycle.

As an illustration, consider an interrupt request on IRQ2 of CNTRL-1. By definition, a request on IRQ2 must have been asserted by CNTRL-2. If IRQ2 wins the priority resolution on CNTRL-1, the interrupt acknowledge cycle returned by the CPU following the interrupt is destined for CNTRL-2, not CNTRL-1. CNTRL-1 will see the INTA# signal, and knowing that the actual destination is CNTRL-2, will broadcast a slave identification code across the internal cascade lines. CNTRL-2 compares this incoming value with the 010b stored in ICW3. Following a positive decode of the incoming message from CNTRL-1, CNTRL-2 drives the appropriate interrupt vector onto the data bus during the second interrupt acknowledge cycle.

Bit	Description
7:3	**Reserved:** Must be 0 when programming this register.
2:0	**Slave Identification Code:** The Slave Identification code must be programmed to 010b during the initialization sequence. The code stored in ICW3 is compared to the incoming slave identification code broadcast by the master controller during interrupt acknowledge cycles.

3.4.3.5 ICW4—Initialization Command Word 4 Register

I/O Address: INT CNTRL-1—021h
INT CNTRL-2—0A1h
Default Value: 01h
Attribute: Write Only
Size: 8 bits

Both interrupt controllers must have ICW4 programmed as part of their initialization sequence. Minimally, the microprocessor mode bit (bit 0) must be set to a 1 to indicate an Intel architecture-based platform. Failure to program this bit will result in improper controller operation during interrupt acknowledge cycles. Additionally, the Automatic End of Interrupt (AEOI) may be selected, as well as the Special Fully Nested Mode (SFNM) of operation.

Bit	Description
7:5	**Reserved:** Must be zero when programming this register.
4	**Special Fully Nested Mode (SFNM):** When bit 4 = 1, the special fully nested mode is enabled. When bit 4 = 0, the special fully nested mode is disabled (this is the normal mode).
3	**Buffered Mode (BUFM):** Must be 0 (non-buffered mode).
2	**Master/Slave in Buffered Mode:** Must be 0.
1	**Automatic End of Interrupt (AEO):** When bit 1 = 1, the automatic end of interrupt mode is selected. When bit 1 = 0, the normal end of interrupt is selected. This bit should normally be programmed to 0.
0	**Microprocessor Mode:** The Microprocessor Mode bit must be programmed to 1 to indicate that the interrupt controller is operating in an Intel Architecture-based system. Never program this bit to 0.

3.4.3.6 OCW1—Operational Control Word 1 Register

I/O Address: INT CNTRL-1—021h
INT CNTRL-2—0A1h
Default Value: 00h
Attribute: Read/Write
Size: 8 bits

OCW1 sets and clears the mask bits in the Interrupt Mask Register (IMR). Each interrupt request line may be selectively masked or unmasked any time after initialization. Note that masking IRQ2 on CNTRL-1 also masks all of controller 2's interrupt requests (IRQ[15:8]). Reading OCW1 returns the controller's mask status.

The IMR stores the bits that mask the interrupt lines to be masked. The IMR operates on the IRR. Masking of a higher priority input does not affect the interrupt request lines of lower priority. Unlike status reads of the ISR and IRR, for reading the IMR, no OCW3 is needed. The output data bus contains the IMR when a read occurs to I/O address 021h or 0A1h (OCW1). All writes to OCW1 must occur following the ICW1-ICW4 initialization sequence, since the same I/O addresses are used for OCW1, ICW2, ICW3 and ICW4.

Bit	Description
7:0	**Interrupt Request Mask (Mask [7:0]):** When a 1 is written to any bit in this register, the corresponding IRQx line is masked. For example, if bit 4 = 1, IRQ4 is masked. Interrupt requests on IRQ4 will not set channel 4's interrupt request register (IRR) bit as long as the channel is masked. When a bit = 0 (default), the corresponding IRQx mask bit is cleared, and interrupt requests are accepted by the controller. Note that masking IRQ2 on CNTRL-1 also masks the interrupt requests from CNTRL-2, which is physically cascaded to IRQ2.

Bit	Interrupt	Bit	Interrupt
0	IRQ0	4	IRQ4
1	IRQ1	5	IRQ5
2	IRQ2	6	IRQ6
3	IRQ3	7	IRQ7

3.4.3.7 OCW2—Operational Control Word 2 Register

I/O Address: INT CNTRL-1—020h
INT CNTRL-2—0A0h
Default Value: Bit[4:0] = undefined, Bit[7:5] = 001
Attribute: Write Only
Size: 8 bits

OCW2 controls both the Rotate Mode and the End of Interrupt Mode, and combinations of the two. OCW2 also selects individual interrupt channels during three of the seven commands. The three low order bits (labeled L2, L1 and L0) are used when bit 6 is set to a 1 during the command. Following a CPURST or ICW initialization, the controller enters the fully nested mode of operation. Non-specific EOI without rotation is the default. Both rotation mode and specific EOI mode are disabled following initialization.

<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>7:5</td><td>Rotate and EOI Codes: These three bits control the Rotate and End of Interrupt modes and combinations of the two as shown below. Bit 7 = R(rotate), Bit 6 = SL, and Bit 5 = EOI.
<table>
<tr><th>Bits[7:5]</th><th>Rotate and EOI Modes</th><th>Bit[7:5]</th><th>Rotate and EOI Modes</th></tr>
<tr><td>001</td><td>Non-Specific EOI Command</td><td>000</td><td>Rotate in Auto EOI Mode (clear)</td></tr>
<tr><td>011</td><td>Specific EOI Command</td><td>111</td><td>Rotate on Specific EOI Command*</td></tr>
<tr><td>101</td><td>Rotate on Non-Specific EOI Command</td><td>110</td><td>Set Priority Command*</td></tr>
<tr><td>100</td><td>Rotate in Auto EOI Mode (set)</td><td>010</td><td>No Operation</td></tr>
</table>
NOTE:
*Interrupt Select Levels are used.</td></tr>
<tr><td>4:3</td><td>OCW2 Select: When selecting OCW2, bits 3 and 4 must both be 0. If bit 4 is a 1, the interrupt controller interprets the write to this port as an ICW1. Therefore, always ensure that these bits are both 0 when writing an OCW2.</td></tr>
<tr><td>2:0</td><td>Interrupt Level Select (L2, L1, L0): L2, L1, and L0 determine the interrupt level acted on when the SL bit is active. A simple binary code, outlined below, selects the channel for the command to act on. When the SL bit is inactive, these bits do not have a defined function. In this case, programming L2, L1 and L0 to 0 is sufficient.
<table>
<tr><th>Bits[2:0]</th><th>Interrupt Level</th><th>Bits[2:0]</th><th>Interrupt Level</th></tr>
<tr><td>000</td><td>IRQ0(8)</td><td>100</td><td>IRQ4(12)</td></tr>
<tr><td>001</td><td>IRQ1(9)</td><td>101</td><td>IRQ5(13)</td></tr>
<tr><td>010</td><td>IRQ2(10)</td><td>110</td><td>IRQ6(14)</td></tr>
<tr><td>011</td><td>IRQ3(11)</td><td>111</td><td>IRQ7(15)</td></tr>
</table></td></tr>
</table>

3.4.3.8 OCW3—Operational Control Word 3 Register

I/O Address: INT CNTRL-1—020h
INT CNTRL-2—0A0h
Default Value: Bit[6,0] = 0, Bit[7,4:2] = undefined, Bit[5,1] = 1
Attribute: Read/Write
Size: 8 bits

OCW3 serves three important functions—Enable Special Mask Mode, Poll Mode control, and IRR/ISR register read control. First, OCW3 is used to set or reset the Special Mask Mode (SMM). The Special Mask Mode can be used by an interrupt service routine to dynamically alter the system priority structure while the routine is executing, through selective enabling/disabling of the other channel's mask bits.

Second, the Poll Mode is enabled when a write to OCW3 is issued with bit 2 equal to 1. The next I/O read to the interrupt controller is treated like an interrupt acknowledge—a binary code representing the highest priority level interrupt request is released onto the bus.

Third, OCW3 provides control for reading the In-Service Register (ISR) and the Interrupt Request Register (IRR). Either the ISR or IRR is selected for reading with a write to OCW3. Bits 0 and 1 carry the encoded command to select either register. The next I/O read to the OCW3 port address returns the register status specified during the previous write. The register specified for a status read is retained by the interrupt controller. Therefore, a write to OCW3 prior to every status read command is unnecessary, provided the status read desired is from the register selected with the last OCW3 write.

Bit	Description
7	**Reserved:** Must be zero when programming this register.
6	**Special Mask Mode (SMM):** If SMME = 1 and SMM = 1 the interrupt controller enters Special Mask Mode. If SMME = 1 and SMM = 0 (default), the interrupt controller is in normal mask mode. When SMME = 0, SMM has no effect.
5	**Special Mask Mode Enable (SMME):** When ESMM = 1 (default), the SMM bit is enabled to set or reset the Special Mask Mode. When ESMM = 0, the SMM bit becomes a ''don't care''.
4:3	**OCW3 Select:** When selecting OCW3, bit 3 must be a 1 and bit 4 must be 0. If bit 4 = 1, the interrupt controller interprets the write to this port as an ICW1. Therefore, always ensure that bits[4:3] = 01 when writing an OCW3.
2	**Poll Mode Command:** When bit 2 = 0, the Poll command is not issued. When bit 2 = 1, the next I/O read to the interrupt controller is treated as an interrupt acknowledge cycle. An encoded byte is driven onto the data bus, representing the highest priority level requesting service.
1:0	**Register Read Command:** Bits[1:0] provide control for reading the In-Service Register (ISR) and the Interrupt Request Register (IRR). When bit 1 = 0, bit 0 does not affect the register read selection. When bit 1 = 1, bit 0 selects the register status returned following an OCW3 read. If bit 0 = 0, the IRR is read. If bit 0 = 1, the ISR is read. Following ICW initialization, the default OCW3 I/O address read is ''read IRR''. To retain the current selection (read ISR or read IRR), always write a 0 to bit 1 when programming this register. The selected register can be read repeatedly without reprogramming OCW3. To select a new status register, OCW3 must be reprogrammed prior to attempting the read.

3.4.3.9 ELCR1—Edge/Level Triggered Register

I/O Address: INT CNTRL-1—4D0h
Default Value: 00h
Attribute: Read/Write
Size: 8 bits

The ELCR1 Register selects either edge triggered or level sensitive operations for the IRQ[7:3] signals. In edge triggered mode, the interrupt is recognized by a low to high transition. In level sensitive mode, the interrupt is recognized by a low level. Note that IRQ0, IRQ1 and IRQ2 are not programmable and are always edge sensitive. The default for IRQ[7:3] is edge triggered.

Bit	Description
7:3	**Bit 7 - Bit 3: IRQ[7:3] ECL:** Bit 7 to bit 3 select edge trigger or level sensitive modes for IRQ[7:3], respectively. When a bit is 1, the corresponding IRQx is in the level sensitive mode and when a bit is 0, the corresponding IRQx is in the edge trigger mode.
2:0	**Reserved:** Must be 0 when programming this register.

3.4.3.10 ELCR2—Edge/Level Triggered Register

I/O Address: INT CNTRL-2—4D1h
Default Value: 00h
Attribute: Read/Write
Size: 8 bits

The ELCR2 Register selects either edge triggered or level sensitive operations for the IRQ[15,14,12:9] signals. In edge triggered mode, the interrupt is recognized by a low to high transition. In level sensitive mode, the interrupt is recognized by a low level. Note that IRQ13 and IRQ8 are not programmable and are always edge sensitive. The default for IRQ[15,14,12:9] is edge triggered.

Bit	Description
7:6	**Bit 7 - Bit 6: IRQ[15,14] ECL:** Bit 7 and bit 6 select edge trigger or level sensitive modes for IRQ[15,14], respectively. When a bit is 1, the corresponding IRQx is in the level sensitive mode and when a bit is 0, the corresponding IRQx is in the edge trigger mode.
5	**Reserved:** Must be 0 when programming this register.
4:1	**Bit 4 - Bit 1: IRQ[12:9] ECL:** Bit 4 to bit 1 select edge trigger or level sensitive modes for IRQ[12:9], respectively. When a bit is 1, the corresponding IRQx is in the level sensitive mode and when a bit is 0, the corresponding IRQx is in the edge trigger mode.
0	**Reserved:** Must be zero when programming this register.

3.4.4 X-BUS REGISTER DESCRIPTION

There are two X-Bus registers described in this section—the Reset X-Bus IRQ[12,1] Register and the Coprocessor Error Register. These registers can be accessed from the CPU, PCI Bus, or ISA Bus.

3.4.4.1 RIRQ—Reset IRQ[12,1]

I/O Address: 60h
Default Value: NA
Attribute: Read only
Size: 8 bits

Address locations 60h and aliased address 62h are used to clear the mouse interrupt (IRQ12) and keyboard interrupt (IRQ1). When the mouse interrupt function is enabled (bit 4 in the X-Bus Chip Select Register is 1), the mouse interrupt function is provided on the IRQ12/M input signal. In this mode, a mouse interrupt generates an interrupt through IRQ12 to the Host CPU. A read of 60h releases IRQ12. If bit 4 = 0 in the X-Bus Chip Select Register, a read of 60h or 62h has no effect on IRQ12/M. Note, however, that a read of these addresses always clears the keyboard interrupt (IRQ1). Reads and writes to this register flow through to the ISA Bus.

Bit	Description
7:2	**Reset IRQ[12,1]:** No specific pattern. A read of address 60h executes the command.
1:0	**Reset IRQ[12,1]:** No specific pattern. A read of address 60h executes the command.

3.4.4.2 CPERR—Coprocessor Error Register

I/O Address: F0h
Default Value: NA
Attribute: Write only
Size: 8 bits

Writes to this address are monitored by the IB. Writing to address F0h causes the IB to drive IGNNE# low to the CPU (informing the CPU to ignore future coprocessor errors). The IB also negates IRQ13 (internal to the IB). Note, that IGNNE# is not asserted unless FERR# is active. Reads and writes to this register flow through to the ISA Bus.

Bit	Description
7:0	**Ignore Coprocessor Error Command:** No special pattern required: A write to address F0h executes the command.

3.4.5 NMI REGISTER DESCRIPTION

An NMI is an interrupt requiring immediate attention and has priority over the normal interrupt lines (IRQx). The IB indicates error conditions by generating a non-maskable interrupt. NMI interrupts (special cycles) are caused by the following conditions:

1. System errors on the PCI Bus. SERR# is driven low by a PCI resource when this error occurs.
2. Parity errors on the add-in memory boards on the ISA expansion bus. IOCHK# is driven low when this error occurs.
3. Main memory parity errors, through the SERR# signal.

There are two 8-bit registers that support NMI—The NMI Status and Control (NMISC) Register and the NMI Enable and Real Time Clock Address (NMIERTC) Register. These registers can be accessed from the CPU, PCI Bus, or ISA Bus. Note that masking the NMI signal for all sources via the NMIERCT Register does not affect the input NMI status conditions (i.e., bits 6 and 7 in the NMISC Register). This means that, if NMI is masked and then unmasked, an NMI will occur if an NMI had previously been detected. To ensure that all NMI requests are serviced, the NMI service routine software flow should be as follows:

1. NMI is detected by the CPU on the rising edge of the NMI input.
2. The CPU reads NMI status via the NMISC Register to determine the NMI source. Software may then set the status bits that caused the NMI to 0. Between the time the CPU reads the NMI sources and sets them to a 0, an NMI may have been generated by another source. In this case, the NMI signal remains asserted. If this happens, the new NMI source will not be recognized by the because there was no edge on NMI.
3. Software must then disable the NMI signals in the NMIERTC Register. This causes the NMI output to transition low then high if there are any pending NMI sources.

3.4.5.1 NMISC—NMI Status and Control Register

I/O Address: 061h
Default Value: 00U0 0000
Attribute: Read/Write
Size: 8 bits

This register provides status of various system components, speaker counter (Counter 2) output control, and gates the counter output that drives the SPKR signal.

Bit		Description
7	RO	**SERR# NMI Source Status:** System agents on the PCI Bus (PCI devices or main memory) assert the SERR# signal to report system errors. When the SERR# signal is asserted (and bit 2 of this register is 0), this bit is set to a 1. In addition, if bit 7 of the NMIERTC Register is 1, an NMI is generated. Software can clear this bit and the interrupt by setting bit 2 to 0 and then setting bit 2 to 1. This bit is read only. When writing to this register, bit 7 must be 0.
6	RO	**IOCHK# NMI Source Status:** Expansion boards on the ISA Bus assert IOCHK# to request high priority servicing (e.g., parity errors on memory cards). When the IOCHK# signal is asserted (and bit 3 of this register is 0), this bit is set to a 1. In addition, if bit 7 of the NMIERTC Register is 1, an NMI is generated. Software can clear this bit and the interrupt by setting bit 3 to 0 and then setting bit 3 to 1. This bit is read only. When writing to this register, bit 6 must be 0.
5	RO	**Timer Counter 2 OUT Status:** This bit reflects the current state of the Interval Timer Counter 2 OUT signal. Counter 2 must be programmed following a CPURST for this bit to have a determinate value. This bit is read only. When writing to this register, bit 5 must be 0.
4	RO	**Refresh Cycle Toggle:** The Refresh Cycle Toggle signal toggles from either 0 to 1 or 1 to 0 following every refresh cycle. This bit is read only. When writing to this register, bit 4 must be 0.
3	R/W	**IOCHK# NMI Enable:** When bit 3 = 1, IOCHK# NMI's are disabled and cleared, and bit 6 of this register is disabled (always reads 0). When bit 3 = 0 (default), bit 6 is enabled and, if bit 7 of the NMIERTC Register is 1, the IOCHK# NMI is enabled.
2	R/W	**PCI SERR# Enable:** When bit 2 = 1, SERR# NMI's are disabled and cleared, and bit 7 of this register is disabled (always reads 0). When bit 2 = 0 (default), bit 7 is enabled and, if bit 7 of the NMIERTC Register is 1, the SERR# NMI is enabled.
1	R/W	**Speaker Data Enable:** This bit enables/disables the SPKR output signal. When bit 1 = 1, the SPKR output signal is enabled and equivalent to the Counter 2 OUT signal. When bit 1 = 0 (default), the SPKR output is disabled and always 0.
0	R/W	**Timer Counter 2 Enable:** When bit 0 = 1, Timer Counter 2 counting is enabled. When bit 0 = 0 (default), Counter 2 counting is disabled.

3.4.5.2 NMIERTC—NMI Enable and Real Time Clock Address Register

I/O Address: 070h
Default Value: Bit[6:0] = undefined, Bit 7 = 1
Attribute: Write Only
Size: 8 bits

This register enables/disables all NMIs and provides a real time clock address pointer field to address memory locations. Do not modify the contents of this register without considering the effects on the state of the other bits. Reads and writes to this register address flow through to the ISA Bus.

Bit	Description
7	**NMI Enable (NMIE):** This bit provides a mask of the NMI output signal. When bit 7 = 1 (default), NMI is disabled for all sources and the NMI signal is negated. When bit 7 = 0, NMIs are enabled. Setting this bit to 1 does not clear or disable the NMI status conditions. Thus, if NMI is disabled then enabled via this register, an NMI will occur if one of the NMI status bits (6 or 7) is set in the NMISC Register.
6:0	**Real time Clock Address:** Used by the Real Time Clock on the Base I/O component to address memory locations.

3.4.6 POWER MANAGEMENT REGISTER DESCRIPTION

This section describes two power management registers—APMS and APMC Registers. These registers are located in normal I/O space and must be accessed (via the CPU or PCI Bus) with 8 bit accesses. Note that the rest of the power management registers are located in PCI configuration space (see Section 3.3, PCI Configuration Registers).

3.4.6.1 APMC—Advanced Power Management Control Port

I/O Address: 0B2h
Default Value: 00h
Attribute: Read/Write
Size: 8 Bits

This register passes data (APM Commands) between the OS and the SMI handler. In addition, writes can generate an SMI and reads can cause STPCLK# to be asserted. The IB operation is not affected by the data in this register.

Bit	Description
7:0	**APM Control Port (APMC):** Writes to this register store data in the APMC Register and reads return the last data written. In addition, writes generate an SMI, if bit 7 of the SMIEN Register and bit 0 of the SMICNTL Register are both is set to 1. Reads cause the STPCLK# signal to be asserted, if bit 1 of the SMICNTL Register is set to 1. Reads do not generate an SMI.

3.4.6.2 APMS—Advanced Power Management Status Port

I/O Address: 0B3h
Default Value: 00h
Attribute: Read/Write
Size: 8 Bits

This register passes status information between the OS and the SMI handler. The IB operation is not affected by the data in this register.

Bit	Description
7:0	**APM Status Port (APMS):** Writes store data in this register and reads return the last data written.

4.0 FUNCTIONAL DESCRIPTION

This section describes the 82420EX PCIset functions and hardware interfaces including the PCIset memory and I/O address map, system arbitration, DMA, interrupt controller, Timer/counter, power management, and clock/reset. The Clock/Reset section also covers the strapping options for the different hardware configuration modes and provides tables listing the state of each 82420EX PCIset output or bi-directional signal during a hard reset. The Host Bus, PCI Bus, ISA Bus, X-Bus, and the PSC/IB Link interfaces are described. The L2 cache and main memory DRAM interfaces and associated memory arrays are covered.

4.1 Memory and I/O Address Map

The 82420EX PCIset interfaces to three system Buses—CPU, PCI, and ISA Buses. The 82425EX PSC provides positive decode for certain I/O and memory space accesses on the CPU and PCI Buses. These decodes include accesses to the PCI Local Bus IDE (CPU only), main memory (CPU, ISA, and PCI), ISA-Compatible registers (CPU, ISA, and PCI), and the PSC's I/O Control Registers (CPU only). In addition, the PSC subtractively decodes certain CPU/PCI cycles.

The 82426EX IB provides positive decode for certain ISA I/O and memory space accesses. These decodes include accesses to the ISA-Compatible registers (for ISA master and DMA initiated cycles), main memory (for ISA and DMA initiated cycles), BIOS, X-Bus, and system events for SMM support. Note that DMA devices and ISA masters can not access the PCI or CPU Buses.

4.1.1 MEMORY ADDRESS MAP

The PSC positively decodes accesses to main memory space (128 MBytes maximum as programmed in the DRB Registers). Accesses in the following ranges are forwarded to the PCI Bus: TOM to 128 MBytes, (2 GByte minus 128 MByte) to (2 GByte plus 128 MByte), and 4 GByte to (4 GByte minus 128 MByte).

All other memory spaces are not intended for use.

NOTE:
The PSC does not decode host CPU address signals A[29:27].

All CPU/PCI accesses to the 0–640 KByte main memory region are forwarded to main memory (Table 10). Accesses to the video region (640–768 KBytes) are subtractively decoded to the ISA Bus. Accesses to the 768 KByte to 1 MByte region are controlled by attribute bits in the PAM Registers. Accesses to the region between 1 MByte and 128 MByte are either sent to main memory or subtractively decoded to either PCI or ISA.

Table 10. CPU/PCI 0 to 128 MByte Address Map

Memory Segment	Decode
16–128 MBytes	Positive decode(1,2)
Top of Hole to 16 MBytes	Positive decode(1,2)
Bottom of Hole to the Top of Hole	Subtractive decode
1 MByte to the Bottom of Hole	Positive decode(1,2)
768 KByte to 1 MByte	Positive decode or subtractively decoded to ISA(3)
640–768 KByte (Video)	Subtractively decoded to ISA
0–640 KByte	Positive decode(4)

NOTES:
1. As programmed in the DRAM Row Boundary Register.
2. For memory accesses that are > top of DRAM <16M, the PSC subtractively decodes and forwards to ISA.
3. The 82420EX allows 13 programmable memory and cacheability attributes on 13 memory segments of various sizes in the ISA compatiblity hole (768 KBytes to 1 MByte). Refer to the PAM Register description.
4. Always in DRAM.

All ISA master and DMA accesses to memory locations 0–640 KByte are forwarded to main memory. ISA memory accesses from 1 MByte to the top of memory are forwarded to main memory, except for accesses to the programmable memory hole. ISA accesses from 768 KByte to 1 MByte (except for E0000–EFFFFh, if forwarding is enabled) and accesses above the top of main memory are confined to the ISA Bus.

Table 11. DMA and ISA Master Accesses to Main Memory

Memory Space	Response
Top of main memory to 128 MByte	Confine to ISA
1 MByte to top of main memory	Forward to main memory(1)
768 KBytes to 1 MByte	Confine to ISA(2)
640–768 KByte (video)	Confine to ISA
0–640 KByte	Forward to main memory

NOTES:
1. Except accesses to programmed memory hole.
2. If bit 6 is 0 in the XBCSA Register and bit 6 is 1 in the PIRQ0 Register, accesses to E0000–EFFFFh are forwarded to main memory.

The 82420EX supports one hole in main memory, as defined by the MEMHOLE Register. CPU accesses in the memory hole are forwarded to the PCI Bus and, if not claimed, forwarded to the ISA Bus. PCI master accesses in the memory hole are subtractively decoded to ISA, if necessary. ISA master accesses are confined to the ISA Bus.

4.1.2 BIOS MEMORY SPACE

The 82420EX PCIset supports 512 KBytes of BIOS space. This includes the normal 128 KByte space plus an additional 384 KByte BIOS space (known as the enlarged BIOS area). All BIOS regions that are not shadowed in main memory are subtractively decoded.

The 128 KByte BIOS memory space is located at 000E0000–000FFFFFh (top of 1 MByte), and is aliased at FFFE0000–FFFFFFFFh (top of 4 GByte). This 128 KByte block is split into two 64 KByte blocks. CPU/PCI accesses to the top 64 KByte region (000F0000–000FFFFFh) that are not claimed by main memory or PCI, are forwarded to ISA. The subsequent ISA cycle always generates a BIOS chip select (asserts BIOSCS#).

CPU/PCI accesses to the bottom 64 KByte region (000E0000–000EFFFFh) that are not claimed by main memory or PCI are forwarded to ISA. The subsequent ISA cycle generates a BIOS chip select, if lower BIOS is enabled (via the XBCSA Register).

The additional 384 KByte region resides at FFF80000–FFFDFFFFh. When enabled (via the XBCSA Register), CPU/PCI memory accesses to this region are subtractively decoded to the ISA Bus and BIOS chip select is generated.

All ISA BIOS accesses within the F0000–FFFFFh region are confined to the ISA Bus, even if BIOS is shadowed in main memory. Accesses to the E0000–EFFFFh region are confined to the ISA Bus, when this BIOS region is enabled (via the XBCSA Register). When the region is disabled, accesses are forwarded to main memory, if forwarding is enabled (via the PREV Register). Note that bit 6 in the XBCSA Register overrides bit 4 in the PREV Register.

4.1.3 VIDEO FRAME BUFFER

The Video Frame Buffer can be mapped in the following ranges:

1. In the standard VGA range.
2. In a defined memory hole. In this case, DRAM size is limited to 128 MByte Memory minus the hole size. For example, if there is a 2 MByte Frame Buffer hole, somewhere between 1 MByte and 16 MByte, then the maximum DRAM size allowed is 128 MByte minus 2 MByte equal 126 MByte.
3. Above Top of Memory, but under 128 MByte. In this case, maximum DRAM size is limited to 128 MByte minus the Frame Buffer Size.
4. Above 128 MByte. There are three non-aliased ranges above 128 MByte, which can be used for Frame Buffer:

   (2 GByte minus 128 MByte) to 2 GByte — When HA[31:27] = 01111

   2 GByte to (2 GByte plus 128 MByte) — When HA[31:27] = 10000

   (4 GByte minus 128 MByte) to (4 GByte minus .5 MByte) — When HA[31:27] = 11111

4.1.4 I/O ACCESSES

The PSC positively decodes access to the I/O control registers, PCI configuration registers, and ISA-Compatible Registers. For details concerning accessing these registers, see Section 3.0, Register Description. In addition, the PSC positively decodes CPU I/O accesses to the IDE ports, when enabled. For IDE port accesses, see Section 4.5, PCI Local Bus IDE.

4.1.5 SMRAM: PROTECTED SMM MEMORY BLOCK

The 82420EX PCIset supports a dedicated 64 KBytes of SMM memory, called SMRAM. The SMRAM is accessible only when certain conditions are met. In normal operations, the SMRAM is hidden. SMRAM can be located at the A0000–F0000h segment. The SMRAM can be enabled/disabled and programmed via the SMRAM Control Register.

When SMRAM is hidden, the whole memory space can be accessed, excluding the SMRAM block. When the SMRAM is visible, most of the memory space is visible, in addition to the SMRAM block. Only the memory block that shares the same bus address ranges with SMRAM cannot be accessed in this case.

SMRAM is visible under the following conditions:

- The CPU is in SMM, performing a memory cycle in the SMRAM range, while the SMRAM is not manually closed. This is indicated by SMIACT# = 0, HLDA = 0, HA is in the SMRAM range (as programmed by SMBASE field of the SMRAMCON Register), and SMCLS = 0. The SMCLS bit only affects data cycles and is ignored for code read cycles.

- The CPU is not in SMM, performing a memory cycle in the SMRAM range, while the SMRAM is manually opened. This is indicated by SMIACT# = 1, HLDA = 0, HA is in the SMRAM range, and the SMOPN = 1.
- On each CPU access when SMIACT# is asserted (or SMOPN = 1), the main memory address is compared to the selected SMM memory address as determined by the SMBASE field. These bits allow the user to select from eight different 64 KByte main memory locations used for SMM memory.

4.2 PSC/IB Link Interface

The PSC and IB communicate using the PSC/IB interface. Interface communications include CPU/PCI accesses of the IB internal registers, CPU/PCI cycles forwarded to the ISA Bus, and ISA master or DMA accesses to main memory. The PSC/IB Link interface is a point-to-point communication connection between the PSC and the IB.

Four sideband signals synchronize data flow and bus ownership—Link Request (LREQ#), Link Grant (LGNT#), Command Valid (CMDV#), and Slave Idle (SIDLE#). LREQ# and LGNT# are used by the IB to arbitrate for link mastership. Only the IB drives LREQ# while only the PSC drives LGNT#. CMDV# is driven by the current link master, while SIDLE# is driven by the current link slave. Commands, addresses, and data are transferred between the PSC and IB using the host address bus signals (A[17:2]).

4.3 Host CPU Interface

The 82420EX PCIset provides a host interface to all of the Intel486 family of processors and upgrades.

4.3.1 HOST BUS SLAVE DEVICE

The PCIset can be configured (via the HOST Device Control Register) to support an Intel486 Host Bus Slave device (specifically, a graphics device). Two special signals (HDEV# and HRDY#) as defined by the VL Bus specification are used in the interface to the Host Bus slave. The PSC can be configured to monitor HDEV# for all memory and I/O ranges that are not positively decoded by the PSC. The PSC can be configured to monitor HRDY# and return RDY# to the CPU, based on HRDY#. The host device may include an I/O range, memory range or both I/O and memory ranges. In all cases, these ranges must not be programmed (positively decoded) by the PSC. The host device's memory ranges are non-cacheable.

NOTES:

1. DMA, ISA Masters and PCI masters cannot access the Host Bus device.
2. The PSC does not contain a time-out mechanism to recover a cycle when HDEV# is asserted but HRDY# is not asserted. When the Host Bus device asserts HDEV#, it is assumed that the HRDY# assertion will follow.
3. Host Bus Device—Read and Write fastest timing. During a CPU read (fastest timing programmed), the Host Bus device must not start driving the data bus until two Host Bus clocks after the active ADS# period. This is required so the L2 cache can drive the data bus for a 0 wait-state L2 read. During a CPU write, the Host Bus device can respond with HRDY# in the cycle following ADS#, to achieve a 0 wait-state write cycle.
4. If the Host bus slave device is implemented on the motherboard, the graphics controller and video DAC chip must be accessed through the host bus. The graphics ROM must also be accessed through the host bus or integrated into system BIOS. If the graphics ROM is not integrated into system BIOS, and the graphics ROM is shadowed into DRAM, the host device must not respond with HDEV# when the graphics ROM area is accessed. If the graphics ROM is not shadowed, the PSC PAM registers must be programmed not to respond to the graphics ROM area.
5. If the host bus slave device is implemented using a connector, the graphics controller and video DAC chip must be accessed through the host bus. The graphics ROM can be accessed through the ISA bus. However, when accessing the ROM BIOS, the card in the connector must latch the address with ADS#.
6. Not all host bus slave devices use all of the host address lines for decode. If this is the case, note that PCI memory and system memory is limited by the number of address lines used by the host bus device for decode (e.g. if [25:2] are only used by the host device, usable system memory is limited to 64 MByte, as the host device will alias anything above 64 MByte).

4.3.2 L1 CACHE SUPPORT

The 82420EX PCIset provides signals that support the CPU's L1 cache. For the S-Series CPUs, the signals are the PCD, KEN#, and EADS# signals. For the D-Series and P24T CPUs, the signals are the KEN#, EADS#, CACHE#, and HITM#. The P24T and the D-Series CPUs include certain signals that are not connected to the PCIset. These signals are fixed to 1 or 0, depending on the system configuration as discussed in Table 12.

4.3.3 SOFT AND HARD RESETS

The 82420EX PCIset generates soft reset (SRESET) and hard resets (PCIRST#, RSTDRV, and CPURST).

Soft Reset

SRESET/INIT is generated under the following conditions:

1. Programming the TRC Register (see TRC Register Description).
2. Keyboard KBDRST#: This signal from the keyboard controller is used to generate a soft reset to the CPU via the PSC's SRESET/INIT signal.
3. Shutdown special cycle: When the CPU executes a shutdown special cycle, SRESET is generated.

Hard Reset

The IB generates a hard reset under two conditions:

1. PWROK; When PWROK is driven low, the IB asserts PCIRST#, RSTDRV, and CPURST. These hard reset signals remain asserted until 2 HCLKIN cycles after the rising edge of PWROK.
2. Programming the TRC Register (see TRC Register description).

Reset Distribution

Figure 4 and Figure 5 show how the hard and soft resets are distributed in the system. The specific implementation depends on the CPU type as shown in the figures.

To ensure that SMI# is not generated during SRESET, an external ''OR'' gate must be used as shown in Figure 6.

Table 12. L1 Cache Signals Not Connected to the PSC

Signal	Description
INV	INV input is tied to 1 for P24T and D-Series processor configuration. When INV = 1, the cache line is invalidated as a result of snoop-hit cycle.
HIT	The CPU asserts HIT to indicate that an Inquire cycle hits a line. The PSC only needs to know when the L1 cache line needs to perform write-back, as a result of Inquire cycle. Since the need to perform write backs is indicated by HITM#, the HIT is not needed in the PSC.
FLUSH#	The PSC does not support L1 hardware flush. Since SMRAM must be configured in a non-cacheable region, there is no need for automatic FLUSH# in a 82420 PCIset-based system. Therefore, FLUSH# can be tied to 1. The PSC recognizes the FLUSH special cycles and responds with RDY# to allow external logic to activate FLUSH# (if desired).
BLEN#	BLEN# is tied to 0 for P24T and D-Series configurations to enable bursted write-back cycles.
WB/WT#	WB/WT# is tied to 1 for P24T and D-Series configurations to set all cache lines in write-back mode. Individual lines can be configured in write-through mode by software only.
EWBE#	P24T External Write Buffer Empty input is used to enforce correct cycle ordering in concurrent systems. In 82420 PCIset-based systems, there is only a single active master at a time. Thus, the PSC does not use the EWBE# signal.
PWT	PWT is used as an indication to cache a line in a write-through mode. The PSC L2 Cache update mode can not be set on a line by line basis and thus PWT is not used.

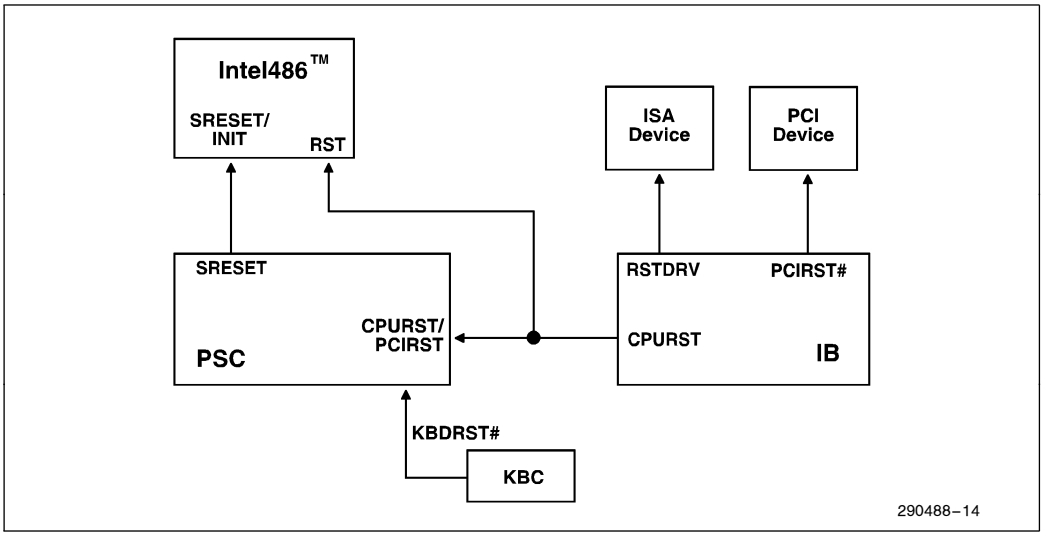


Figure 4. Reset Distribution for CPU's with Hard Reset and Soft Reset Inputs

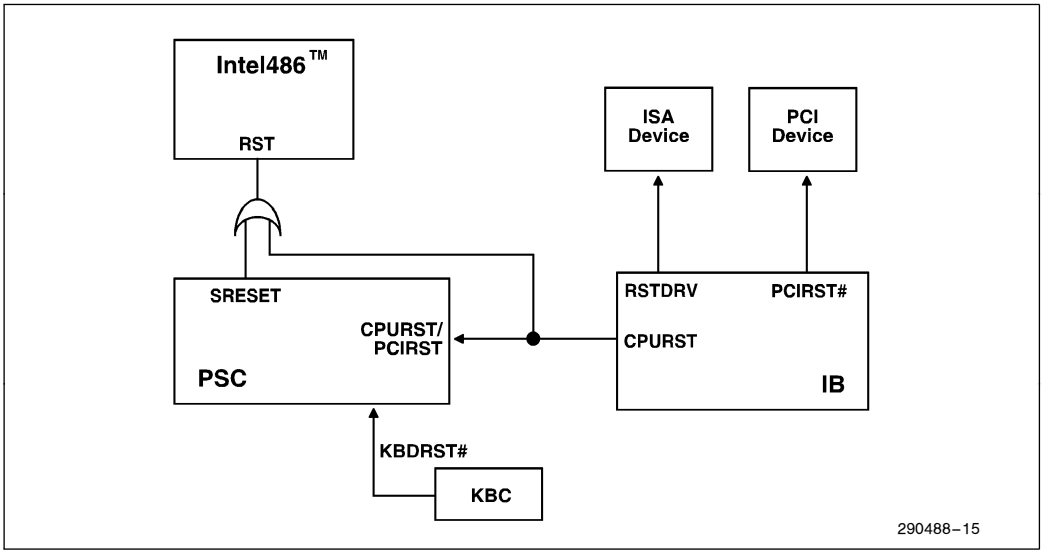


Figure 5. Reset Distribution for CPU's with One Reset Input

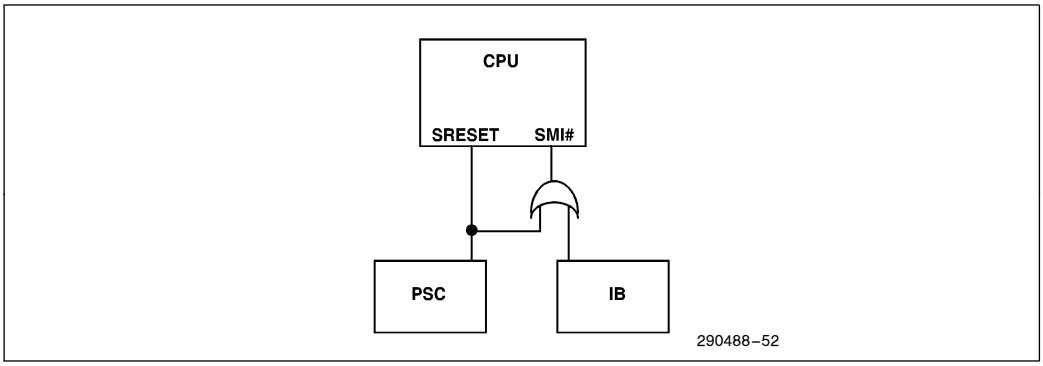


Figure 6. SMI# Gated with SRESET

4.3.4 KEYBOARD CONTROLLER (A20)

The 82420EX supports the generation of A20M# via the keyboard controller. To support the generation of A20M#, external logic may be required (Figure 7), depending on system requirements. ''OR'' gates 1 and 2 ensure that A20M# is negated during a SRESET or CPURST, respectively. ''OR'' gate 2 is not required if the KBC firmware forces KBCA20M# high during a hard reset. ''OR'' gate 3 and the inverter ensure that A20M# is negated when SMIACT# is asserted. ''OR'' gate 3 and the inverter are not required if the SMRAM is located under 1 MByte or at an even 1 MByte boundary and the SMRAM code does not need to go above the 1 MByte range while in SMM mode.

4.4 PCI Interface

The PSC has a standard master/slave PCI Bus interface. As a PCI device, the PSC can be either a master initiating a PCI Bus operation or a target responding to a PCI Bus operation. The PSC is a PCI Bus master for Host-to-PCI accesses and a target for PCI-to-Main memory accesses (or accesses that are forwarded to the ISA Bus). The Host can read or write configuration spaces, PCI memory space, and PCI I/O space.

NOTES:

1. PCI-to-Host accesses are not permitted. However, PCI-to-Main memory cycles that require the L1/L2 caches to be snooped, do invoke Host Bus cycles.
2. ISA-to-PCI accesses are not permitted.

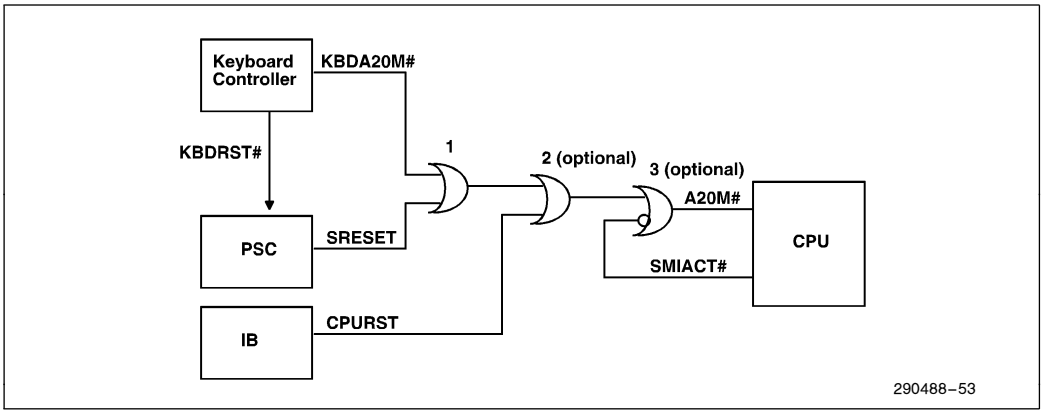


Figure 7. System Connection for Keyboard A20M# Generation

4.4.1 PCI BUS CYCLES SUPPORT

When the host initiates a bus cycle to a PCI device, the PSC becomes a PCI Bus master and translates the CPU cycle into the appropriate PCI Bus cycle. Post buffers permit the CPU to complete Host-to-PCI writes in zero wait-states.

When a PCI Bus master initiates a main memory access, the PSC becomes the target of the PCI Bus cycle and responds to the read/write access. As a PCI master, the PSC generates address parity for read and write cycles, and data parity for write cycles. As a target, the PSC generates data parity for read cycles. During PCI-to-Main memory accesses, the PSC automatically performs cache snoop operations on the Host Bus, if needed, to maintain data consistency.

PCI Bus commands indicate to the target the type of transaction desired by the master. These commands are presented on the C/BE[3:0]# signals during the address phase of a transfer. Table 13 summarizes The PSC's support of the PCI Bus commands.

PSC Supports Other PCI Bridges

The PCI Bus specification supports bridges that connect the system's local PCI Bus with other remote busses (PCI or others). The PSC supports the ability to connect bus bridges onto the local PCI Bus.

One type of PCI bridge interfaces the local PCI Bus to a set of slave (only) devices. In this case, the bridge performs protocol translation and may include write buffers (pointing away from the local PCI). An example of such a bridge is the PCI-to-PCMCIA bridge device (PPEC).

A second type of PCI bridge interfaces the local PCI Bus with another bus that supports masters and slaves—a remote PCI Bus. This type of bridge can generally include write buffers (and pre-fetchers) that are pointing in both directions (to local PCI Bus and away from local PCI Bus).

Table 13. Supported PCI Bus Commands

C/BE[3:0]#	Command Type	Supported As Target	Supported As Master
0000	Interrupt Acknowledge	No	No
0001	Special Cycle	No	No
0010	I/O Read	Yes	Yes
0011	I/O Write	Yes	Yes
0100	Reserved	—	—
0101	Reserved	—	—
0110	Memory Read	Yes	Yes
0111	Memory Write	Yes	Yes
1000	Reserved	—	—
1001	Reserved	—	—
1010	Configuration Read	No	Yes
1011	Configuration Write	No	Yes
1100	Memory Read Multiple	Yes(1)	No
1101	Dual Address Cycle	No	No
1110	Memory Read Line	Yes(1)	No
1111	Memory Write and Invalidate	Yes(2)	No

NOTES:
1. As a target, the PSC treats this command as a memory read command.
2. As a target, the PSC treats this command as a memory write command.

The PSC supports PCI-to-PCI bridges, with the following restrictions:

- The 82420EX PCIset does not allow more than a single active master in the entire system. This restriction prevents a remote PCI Bus master from performing an exclusive access that is claimed by the bridge (the target is on the local PCI Bus), while there is another active master in the system (that may be performing another exclusive access on the local PCI Bus).
- When a master is granted, it is guaranteed that the PSC's PCI write buffers are empty. Since the PSC does not know the status of other bridges's buffers (that point to the PCI) while it grants the CPU, the other bridge's buffers must be disabled.

4.4.2 HOST TO PCI CYCLES

Host bus accesses to PCI Bus are always in the Host Bus address range, as defined by A[31:30,26:2] and the four BE lines. The PCI address lines are driven during the address phase. AD[29:27] lines are driven to the value of A[30], during Host accesses to PCI.

The PSC has the ability to burst up to 32 back-to-back CPU memory writes on the PCI Bus. This function is controlled by the PCICON Register. The PSC is capable of merging 8/16-bit graphic write cycles to the same dword address into the same posted write buffer location (controlled by the PCICON Register). The merged data is then driven as a single dword cycle on the PCI Bus. Byte merging is performed in the compatible VGA range only.

4.4.3 PCI CYCLE TERMINATIONS

The PSC performs a master abort, received target abort, signaled target abort, and a disconnect (as either a master or slave) as described in this section.

Master Abort—PSC as a Master

The PSC performs two types of master abort when a PCI cycle is not claimed by PCI Bus devices.

Master-Abort of Type 1 is performed by the PSC for the following conditions:

- When the memory address is lower than 16 MBytes.
- When the memory address is higher than 16 MBytes, but it is an enabled BIOS range.
- When the I/O address is lower than 64 KBytes.

Type 1 master abort actions:

- Master abort is performed.
- The cycle is forwarded to the ISA Bus.

Master abort of Type 2 is performed by the PSC for the following conditions:

- When the memory address is higher than 16 Mbytes, and it is not an enabled BIOS range.
- I/O address is above 64 KBytes.
- When a configuration access to a PCI device is not claimed.

Type 2 master abort actions:

- Master abort is performed.
- Master abort status bit (DS Register) is set.
- For reads, data of all 1's is returned to the CPU.
- For writes, RDY# is activated to complete the CPU cycle.

Received Target Abort—PSC as a Master:

When a PSC driven cycle is target aborted, the PSC sets the Received Target Abort status bit to 1 (in the DS Register). In addition, when SERR# is enabled, this signal is asserted for a single PCICLK. RDY# is asserted to complete the CPU cycle.

NOTE:
When the CPU attempts an access configuration registers and the function number is not 000, data of all 1's is returned (if it is a read cycle) and Target Abort Status bit is set.

Disconnect—PSC as a Master

When the PSC, as a PCI master, generates a burst memory write, it can be disconnected by the PCI target. The PSC will retry the disconnected cycle before any arbitration changes can be performed, since the PSC write buffers must be emptied and the on-going CPU access must be completed before an arbitration transfer can take place.

Disconnect—PSC as a Target

The PSC, as a PCI target, performs a disconnect when burst PCI master accesses are destined to the ISA Bus. The disconnect is performed at the completion of the first data phase. In addition, for burst PCI master cycles to main memory, the PSC performs a disconnect at the completion of the last data phase in a line boundary.

4.4.4 EXCLUSIVE CYCLES

The PSC, as a PCI master, never performs LOCKed cycles. The CPU does not return active HLDA while it is performing a LOCKed sequence. Also, the CPU is the only active master, as long as HLDA is inactive. Thus, the PSC does not need to drive LOCK to guarantee the CPU atomic LOCK sequence. Note that the 82420EX PCIset supports a bus locking mechanism (i.e., when a PCI master performs locked accesses, the arbitration is not changed, until the locked sequence is completed).

4.4.5 Parity Support

As a master, the PSC generates address parity for read and write cycles, and data parity for write cycles. As a slave, the PSC generates data parity for read cycles. Even parity is generated using the PAR line in the PCICLK following the PCI address or data phase.

The PSC does not check parity or generate SERR#, based on the PCI parity. The PSC only generates SERR# (if enabled via the PCICOM Register), when a main memory read results in a parity error. When a main memory parity error is detected, the PSC activates SERR#, if enabled, for a single PCICLK.

When a main memory parity error is detected and SERR# generation is enabled, the MMPERR bit in the DS Register is set to 1. When SERR# is activated, the SERRS bit in the DS Register is set to 1.

4.5 PCI Local Bus Ide

The PSC has a full-function PCI Local Bus IDE Controller capable of generating high speed PCI Local Bus IDE cycles. The PCI IDE address, control, and data signals are multiplexed with the PCI AD signals (Figure 8). They are buffered by external TTL devices to drive the IDE connector. Only CPU accesses to IDE can use the PCI local bus path. PCI masters and ISA masters can not access the drive connected to the PCI Local Bus.

The PSC's IDE interface supports one IDE connector (two drives). An additional IDE connector could be connected to the ISA or PCI Bus. The PCI IDE interface can be programmed at either the primary address (1F0h–1F7h, 3F6h, 3F7h) or secondary address (170h–177h, 376h, 377h) locations.

The selected IDE's data port, as well as the control/status ports, are accessed through the PCI Local Bus path. The PSC provides data steering to route data between the IDE data bus and the correct Host Bus byte lane. However, the PSC does not support multiple assembly/disassembly cycles for data size mismatches. Data size matching is guaranteed by the IDE device driver. The PSC assumes that all data port accesses (1F0h, 170h) are 16 bits wide. If an 8-bit IDE drive is accessed, the PSC still drives 16 bits onto the PCI AD signals for data port writes to IDE, and drive HD(15:0) for data port reads. Accesses to the PCI Local Bus control and status ports are assumed to be 8-bit accesses and the PSC steers the data to the appropriate byte lane. Table 14 shows the I/O addresses for the various IDE data, control, and status ports:

The PSC controls the timing of the high speed accesses to the PCI IDE connector and provides programmable timing fields (via the LBIDE Register). This allows the PCI local bus IDE timing to be programmed to cover 25 MHz and 33 MHz PCI frequencies, and IDE Modes 1, 2, and 3. The programmable timing also allows additional flexibility in the event that still faster IDE modes are defined in the future. Note that, the faster timing applies to data port accesses only. Accesses to all other ports, except the data port, run with compatible timings.

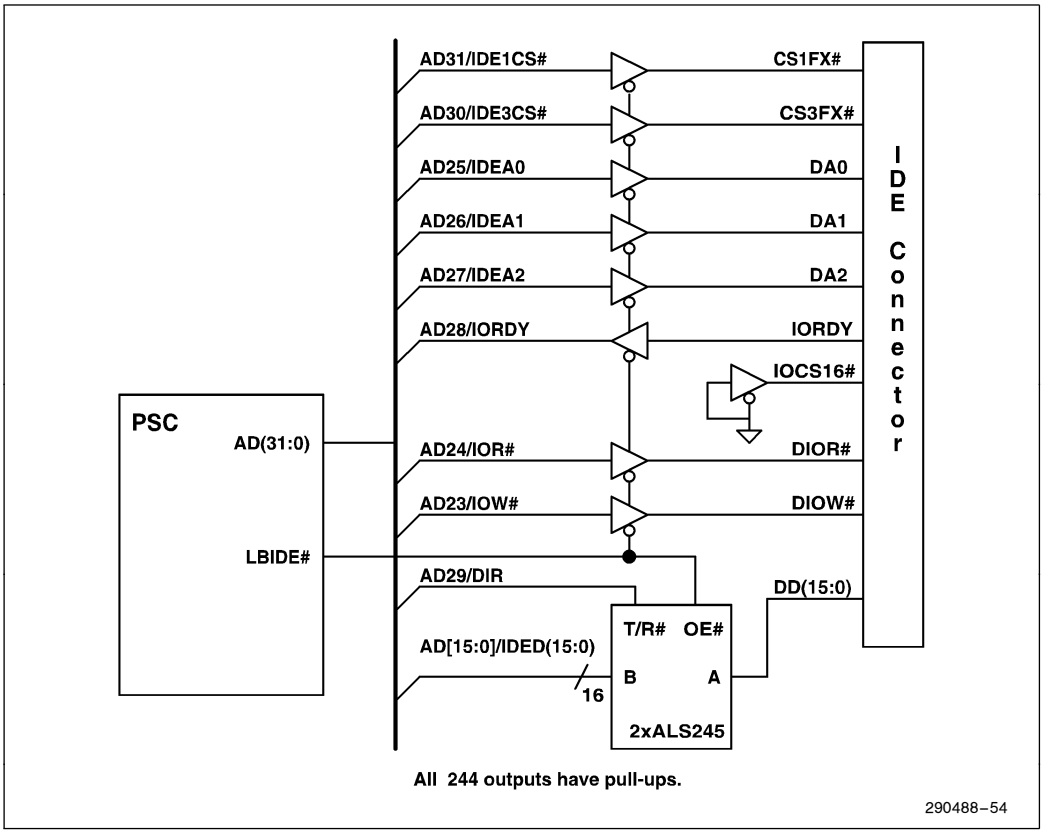


Figure 8. PSC PCI Local Bus IDE Connection

The PSC supports one connector that can be assigned at the primary or secondary address. This one connector can support two drives. The drive is selected through the Drive, Head port at I/O address 1x6h. The CPU writes bit 4 to a 0 to select drive 0, and writes bit 4 to a 1 to select drive 1. The PSC snoops writes to the enabled 1x6h (1F6h for primary, 176h for secondary) and keeps its own copy of bit 4 of I/O 1x6h. The fast timing bank can be programmed to apply to data port accesses to either drive 0, drive 1, or both. Accesses to the non-selected drive run with compatible timings.

Typically in a PC, when reading from port 3x7h, bits[6:0] are provided by the IDE drive and bit 7 is provided by the floppy disk controller as a reflection of the DSKCHG from the floppy disk drive. This occurs for both the primary and secondary locations. The PSC handles CPU I/O reads to port 3x7h in a unique fashion. For example, when the primary address range is enabled, the PSC splits the read to 3F7h and generates both a PSC/IB link interface bus cycle as well as a PCI Local Bus IDE cycle. The PSC takes bit 7 from the link cycle, merges it with bits[6:0] from the PCI local bus IDE cycle, and returns the complete 8 bits to the CPU. If the primary address range is not enabled, only the PSC/IB link interface bus cycle is generated. The same operation applies to 377h reads, when the secondary address range is enabled. This feature permits the PCI Local Bus IDE to be used in a system where, for example, the AIP is placed on an add-in card.

Table 14. IDE I/O Addresses

I/O Address	Port Function (R/W)
1x0h	Data
1x1h	Error/Features
1x2h	Sector Count
1x3h	Sector Number
1x4h	Cylinder Low
1x5h	Cylinder High
1x6h	Drive, Head
1x7h	Status/Command
3x6h	Alternate Status/Device Control
3x7h	Drive Address

NOTE:
x = F for Primary and 7 for Secondary

4.6 ISA Interface

The IB incorporates a fully ISA Bus compatible master and slave interface. The IB directly drives five ISA slots without external data or address buffers. The ISA interface also provides byte swap logic, I/O recovery support, wait-state generation, and SYSCLK generation. The ISA interface supports the following cycle types:

- CPU or PCI master initiated I/O and memory cycles to the ISA Bus.
- DMA compatible cycles between main memory and ISA I/O and between ISA I/O and ISA memory.
- ISA refresh cycles initiated by either the IB or an external ISA master.
- ISA master-initiated memory cycles to main memory and ISA master-initiated I/O cycles to the internal IB registers.

NOTES:

1. The IB does not grant the ISA Bus to an ISA master before gaining ownership of the system (i.e. Host and PCI Buses).
2. All cycles forwarded to main memory run as 16-bit extended cycles (i.e. IOCHRDY is negated until the cycle completes). Because the ISA Bus size is different from the main memory bus size, the data steering logic inside the IB steers the data to the correct byte lanes.

I/O Recovery Support

The I/O recovery mechanism in the IB is used to add additional recovery delay between the CPU or PCI master initiated 8-bit and 16-bit I/O cycles to the ISA Bus. The IB automatically forces a minimum delay of 3.5 SYSCLKs between back-to-back 8- and 16-bit I/O cycles to the ISA Bus. This delay is measured from the rising edge of the I/O command (IOR# or IOW#) to the falling edge of the next I/O command. If a delay of greater than 3.5 SYSCLKs is required, the ISA I/O Recovery Timer Register can be programmed to increase the delay in increments of SYSCLKs. No additional delay is inserted for back-to-back I/O sub-cycles generated as a result of byte assembly or disassembly.

SYSCLK Generation

The IB generates the ISA system clock (SYSCLK). SYSCLK is a divided down version of HCLKOUT and has a frequency of either 8.00 or 8.33 MHz, depending on the HCLKOUT frequency. The clock divisor value is determined by strapping options as discussed in the Clock section.

For CPU or PCI initiated cycles to the ISA Bus, SYSCLK is stretched to synchronize BALE falling to the rising edge of SYSCLK. During CPU or PCI initiated cycles to the IB, BALE is normally driven high, synchronized to the rising edge of SYSCLK and then driven low to initiate the cycle on the ISA Bus. However, if the cycle is aborted, BALE remains high and is not driven low until the next cycle to the ISA Bus.

Data Byte Swapping (ISA Master or DMA to ISA Device)

The data swap logic is integrated in the IB. For slaves that reside on the ISA Bus, data swapping is performed if the slave (I/O or memory) and ISA Bus master (or DMA) sizes differ and the upper (odd) byte of data is being accessed. The data swapping direction is determined by the cycle type (read or write). Table 15 shows when data swapping is provided during DMA and ISA master cycles to ISA slaves.

Table 15. DMA Data Swap

DMA I/O Device Size	ISA Memory Slave Size	Swap	Comments (I/O) ⟷ Memory
8-bit	8-bit	No	SD[7:0] ⟷ SD[7:0]
8-bit	16-bit	No	SD[7:0] ⟷ SD[7:0]
8-bit	16-bit	Yes	SD[7:0] ⟷ SD[15:8]
16-bit	8-bit	No	Not Supported
16-bit	16-bit	No	SD[15:0] ⟷ SD[15:0]

Table 16. 16-bit Master to 8-bit Slave Data Swap

SBHE#	SA0	SD[15:8]	SD[7:0]	Comments
0	0	Odd	Even	Word Transfer (data swapping not required)
0	1	Odd	Odd	Byte Swap(1, 2)
1	0	—	Even	Byte Transfer (data swapping not required)
1	1	—	—	Not Allowed

NOTES:
1. For ISA master read cycles, the IB swaps the data from the lower byte to the upper byte.
2. For ISA master write cycles, the IB swaps the data from the upper byte to the lower byte.

Wait-State Generation

The IB adds wait-states to the following cycles, if IOCHRDY is sampled negated (low). Wait-states are added as long as IOCHRDY remains low.

- During Refresh and IB master cycles (not including DMA) to the ISA Bus.
- During DMA compatible transfers between ISA I/O and ISA memory only.

For ISA master cycles targeted for the IB's internal registers or main memory, the IB always extends the cycle by driving IOCHRDY low until the transaction is complete.

Cycle Shortening

The IB shortens the following cycles, if ZEROWS# is sampled asserted (low).

- During IB master cycles (not including DMA) to 8-bit and 16-bit ISA memory.
- During IB master cycles (not including DMA) to 8-bit ISA I/O only.

For ISA master cycles targeted for the IB's internal registers or main memory, the IB does not assert ZEROWS#. If IOCHRDY and ZEROWS# are sampled low at the same time, IOCHRDY will take precedence and wait-states will be added.

4.7 X-Bus

The 82420EX PCIset provides the decode (chip selects) and X-Bus buffer control (XBUSOE# and XBUSTR#) for a Real Time Clock, Keyboard Controller, and BIOS (Figure 9). The chip selects are generated combinatorially from the ISA SA[16:0] and LA[23:17] address bus. (Note that the ISA master must drive SA[19:16] and LA[23:17] low when accessing I/O space.) The IB also provides PS/2 mouse support via the IRQ12/M signal and coprocessor functions (FERR# and IGNNE#). The chip selects and X-Bus buffer control lines can be enabled/disabled via the XBCSA Configuration Register.

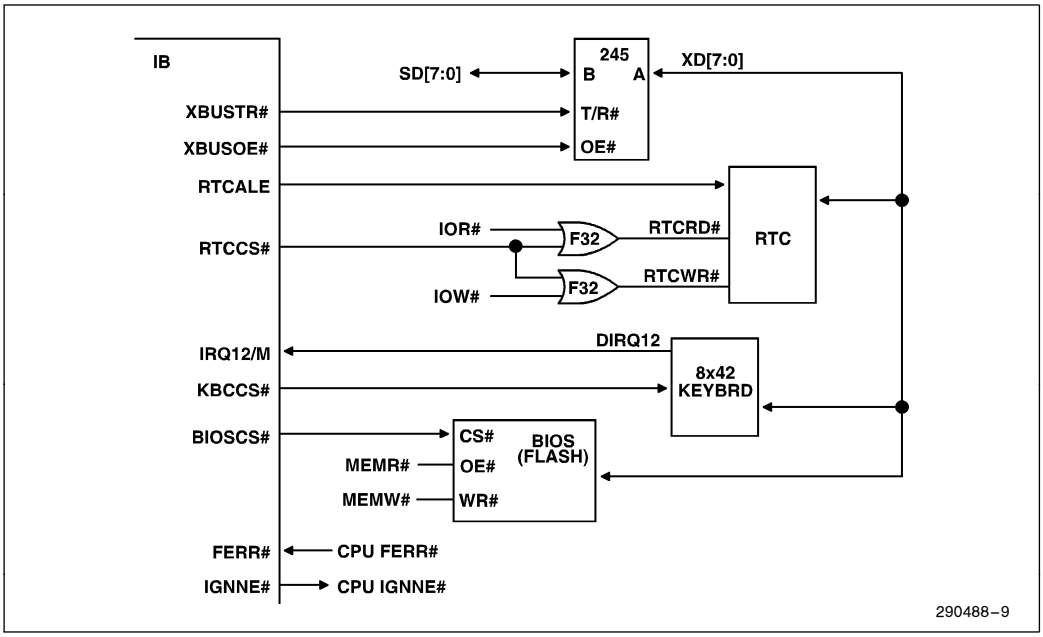


Figure 9. X-Bus Support

4.7.1 COPROCESSOR ERROR FUNCTION

The IB provides coprocessor error support for the CPU (enabled/disabled via the XBCSA Register). FERR# is tied directly to the coprocessor error signal of the CPU. If FERR# is asserted, an internal IRQ13 is generated and the INTR signal is asserted. When a write to I/O location F0h is detected, the IB negates IRQ13 (internal to the IB) and asserts IGNNE#. IGNNE# remains asserted until FERR# is negated. Note, that IGNNE# is not asserted unless FERR# is asserted.

4.7.2 MOUSE FUNCTION

The IB provides a mouse interrupt function (enabled/disabled via the XBCSA Register) on the IRQ12/M input signal. In this mode, a mouse interrupt generates an interrupt through IRQ12 to the Host CPU. The IB informs the CPU of this interrupt via a INTR. A read of 60h or 62h releases IRQ12. If bit 4 = 0 in the XBCSA Register, a read of address 60h or 62h has no effect on IRQ12/M. Reads and writes to this register flow through to the ISA Bus. For additional information, see the IRQ12/M description in the Signal Description.

4.8 System Arbitration

The 82420EX PCIset provides bus arbitration on the Host Bus, PCI Bus, and the PCI/IB Interface (to the ISA Bus). A device that is the master on any bus is the master of the entire system. (i.e., concurrency of more than one active master is not supported).

Signals associated with the system arbitration are the HOLD/HLDA signals (CPU Bus), PREQ[1:0]/PGNT[1:0]# signals (PCI Bus), and the LREQ#/LGNT# signals (PSC/IB Link Interface).

4.8.1 SYSTEM ARBITRATION SCHEME

When there are no active requests, the CPU owns the system. The system arbitration rotates between the PCI Bus, CPU Bus, and Link Interface Bus (on behalf of DMA and ISA Master devices), with the CPU permitted access every other transition.

NOTES:

1. The PSC, as a PCI master, never performs locked cycles. However, locked cycles are supported for PCI masters. When a PCI master performs a locked access, the arbitration is not changed until the locked sequence is completed.
2. After PGNT[1:0]# is asserted by the PSC, it is negated when FRAME# is sampled active (regardless the state of PREQ[1:0]#). The PCI master is expected to continue its current cycle (with potential multiple data phases), and then get-off the PCI Bus. The PSC does not release HOLD until the PCI Bus is idle. When a PCI master is re-tried by the PCI target, PGNT[1:0]# is already negated. Thus, the PCI master must get-off the bus. Since the PSC always gives the bus back to the CPU and the arbitration is rotated, PREQ[1:0]# can remain active as long as the PCI master has cycles to perform.
3. The PSC precludes fast back-to-back PCI master transactions. In addition, the PSC, as a PCI master, does not support fast back-to-back transactions.
4. When the PSC, as a PCI master, is re-tried, target-aborted or master-aborted, by a PCI target, the arbitration mechanism does not assert PGNT[1:0]# or LGNT#.

4.9 DMA Controller

The DMA circuitry incorporates the functionality of two 82C37 DMA controllers with seven independently programmable channels (Channels [3:0] and Channels [7:5]). The DMA supports 8/16-bit device size using ISA-compatible timings and 27-bit addressing as an extension of the ISA-compatible specification. The DMA channels can be programmed for either fixed (default) or rotating priority. The DMA controller also generates ISA refresh cycles. DMA Channel 4 is used to cascade the two controllers and default to cascade mode in the DMA Channel Mode (DCM) Register (Figure 10). In addition to accepting requests from DMA slaves, the DMA controller also responds to requests that are initiated by software. Software may initiate a DMA service request by setting any bit in the DMA Channel Request Register to a 1. The DMA controller for Channels [3:0] is referred to as ''DMA-1'' and the controller for Channels [7:4] is referred to as ''DMA-2''.

Each DMA channel is hardwired to the compatible settings for DMA device sizechannels [3:0] are hardwired to 8-bit, count-by-bytes transfers and channels [7:5] are hardwired to 16-bit, count-by-words (address shifted) transfers. The IB provides the timing control and data size translation necessary for the DMA transfer between the memory (ISA or main memory) and the ISA Bus I/O. ISA-Compatible DMA timing is supported. The DMA controller also features refresh address generation and auto-initialization following a DMA termination.

Note that a DMA device (I/O device) is always on the ISA Bus, but the memory referenced is located on either an ISA Bus device or in main memory. When the IB is running a DMA cycle, it drives the MEMR# or MEMW# strobes, if the address is less than 16 MBytes (000000–FFFFFFh). The IB always generates ISA-Compatible DMA memory cycles. The SMEMR# and SMEMW# are generated if the address is less than 1 MByte (0000000–00FFFFFh). To avoid aliasing problems when the address is greater than 16 MBytes (1000000–7FFFFFFh), the MEMR# or MEMW# strobe is not generated.

The channels can be programmed for any of four transfer modessingle, block, demand, or cascade. Each of the three active transfer modes (single, block, and demand), can perform three different types of transfers (read, write, or verify). Note that memory-to-memory transfers are not supported by the IB. The DMA supports fixed and rotating channel priorities.

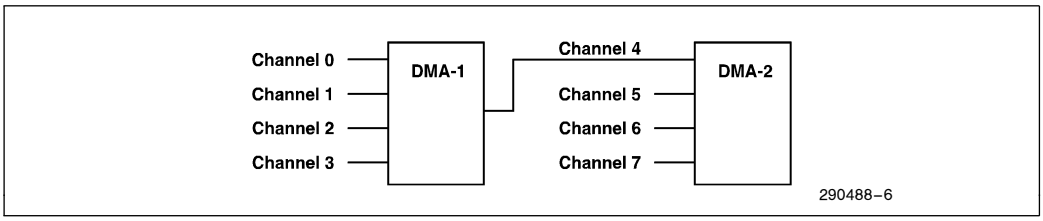


Figure 10. Internal DMA Controller

4.10 Interval Timer

The 82420EX PCIset contains three counters that are equivalent to those found in the 82C54 programmable interval timer. Each counter output provides a key system function. Counter 0 is connected to interrupt controller IRQ0 and provides a system timer interrupt for a time-of-day, diskette time-out, or other system timing functions. Counter 1 generates a refresh request signal and Counter 2 generates the tone for the speaker.

Counter 0 (System Timer)

This counter functions as the system timer by controlling the state of IRQ0 and is typically programmed for Mode 3 operation. The counter produces a square wave with a period equal to the product of the counter period (838 ns) and the initial count value. The counter loads the initial count value one counter period after software writes the count value to the counter I/O address. The counter initially asserts IRQ0 and decrements the count value by two each counter period. The counter negates IRQ0 when the count value reaches 0. It then re-loads the initial count value and again decrements the initial count value by two each counter period. The counter then asserts IRQ0 when the count value reaches 0, re-loads the initial count value, and repeats the cycle, alternately asserting and negating IRQ0.

Counter 1 (Refresh Request Signal)

This counter provides the refresh request signal and is typically programmed for Mode 2 operation. The counter negates refresh request for one counter period (838 ns) during each count cycle. The initial count value is loaded one counter period after being written to the counter I/O address. The counter initially asserts refresh request, and negates it for 1 counter period when the count value reaches 1. The counter then asserts refresh request and continues counting from the initial count value.

Counter 2 (Speaker Tone)

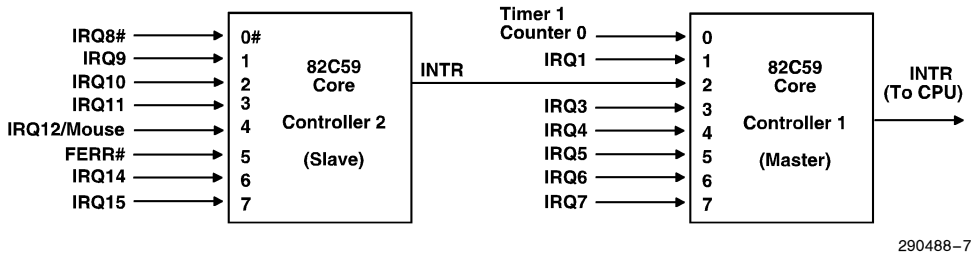
This counter provides the speaker tone and is typically programmed for Mode 3 operation. The counter provides a speaker frequency equal to the counter clock frequency (1.193 MHz) divided by the initial count value. The speaker must be enabled by a write to I/O address 061h.

4.11 Interrupt Controller

The 82420EX PCIset contains an ISA-compatible interrupt controller that incorporates the functionality of two 82C59 interrupt controllers. The two controllers (CNTRL-1and CNTRL-2) are cascaded allowing thirteen external and three internal interrupts (Figure 11). CNTRL-1 and CNTRL-2 are initialized separately and can be programmed to operate in different modes. CNTRL-1 is connected as the master interrupt controller and CNTRL-2 is connected as the slave interrupt controller. The three internal interrupts are used for internal functions only and are not available to the user. IRQ2 cascades the two controllers. IRQ0 provides a system timer interrupt and is tied to the Interval Timer, Counter 0. IRQ13 is connected internally to FERR# for coprocessor error support. The remaining thirteen interrupt lines (IRQ[15,14,12:9,8#,7:3,1]) are available for external system interrupts. Edge or level sense selection is programmable on an individual channel by channel basis. Interrupt steering permits two programmable interrupts (PIRQ0# and PIRQ1#) to be internally routed (steered) to one of eleven interrupts (IRQ[15,14,12:9,7:3]).

NOTES:

1. The standard external IRQ12 signal function or internally generated IRQ12/Mouse function are selected via the XBCSA Register.
2. The IB translates the CPU generated interrupt acknowledge cycle internally into the two INTA# pulses expected by the interrupt controller system.



NOTE:
IRQ0, IRQ2, and IRQ13 are internally connected and do not appear externally.

Figure 11. Block Diagram of the Interrupt Controller

4.11.1 PROGRAMMING THE INTERRUPT CONTROLLER

The Interrupt Controller accepts two types of command words generated by the CPU or bus master—Initialization Command Word (ICWx) and Operation Command Word (OCWx).

Initialization Command Words (ICWs)

Before normal operation can begin, each interrupt controller in the system must be initialized. In the 82C59, this is a two to four byte sequence. However, for the 82420EX PCIset, each controller must be initialized with a four byte sequence. This four byte sequence is required to configure the interrupt controller correctly for the IB implementation. This implementation is ISA compatible.

The four initialization command words are referred to by their acronyms: ICW1, ICW2, ICW3, and ICW4. These command registers are discussed in Section 3.0, Register Description. The sequence must be executed for CNTRL-1 and CNTRL-2. ICW1, ICW2, ICW3, and ICW4 must be written in order. Any divergence from this sequence, such as an attempt to program an OCW, will result in improper initialization of the interrupt controller and unexpected, erratic system behavior. It is suggested that CNTRL-2 be initialized first, followed by CNTRL-1.

Operation Command Words (OCWs):

These are the command words which dynamically reprogram the Interrupt Controller to operate in various interrupt modes. OCW1 masks interrupt lines. OCW2 controls rotation in interrupt rotation priority mode and the End of Interrupt (EOI). OCW3 sets up reads of the ISR and IRR, enables/disables the Special Mask Mode (SMM), and sets up the polled interrupt mode. The OCWs can be invoked any time after initialization.

4.11.2 EDGE AND LEVEL INTERRUPT TRIGGERED MODE

In ISA systems, this mode is programmed using bit 3 in ICW1. For the IB, this bit is disabled and the Edge/Level Control Registers (ELCR1 and ELCR2) are included that select edge and level triggered mode per interrupt input. The default programming is equivalent to programming the LTIM bit (ICW1 bit 3) to a 0 (all interrupts selected for edge triggered mode). Note, that IRQ[13,8#,2,1,0] can not be programmed for level sensitive mode.

In both the edge and level triggered modes, the IRQx input must remain active until after the falling edge of the first INTA#. If IRQx is negated before this time, a default IRQ7 occurs when the CPU acknowledges the interrupt. This can be a useful safeguard for detecting interrupts caused by spurious noise glitches on the IRQ inputs. To implement this feature, the IRQ7 routine is used for ''clean up'' by simply executing a return instruction, thus ignoring the interrupt. If IRQ7 is needed for other purposes, a default IRQ7 can still be detected by reading the ISR. A normal IRQ7 interrupt sets the corresponding ISR bit; a default IRQ7 does not set this bit. If a default IRQ7 routine occurs during a normal IRQ7 routine, however, the ISR remains set. In this case, it is necessary to keep track of whether or not the IRQ7 routine was previously entered. If another IRQ7 occurs, it is a default.

4.11.3 INTERRUPT STEERING

The IB contains two programmable interrupts (PIRQ0 and PIRQ1#) that can be internally routed to one of eleven interrupts (IRQ[15,14,12:9,7:3]) by programming the PIRQx Route Control Registers. One or both PIRQx# lines can be routed to the same IRQx input or interrupt steering can be disabled.

The PIRQx# lines are defined as active low, level sensitive to allow multiple interrupts on a PCI board to share a single line across the connector. When a PIRQx# is routed to a specified IRQ line, the software must change the IRQ's corresponding ELCR bit to level sensitive mode. Note, that this means that the selected IRQ can no longer be used by an ISA device, unless that ISA device can respond as an active low level sensitive interrupt.

4.12 L2 Cache

The L2 cache memory array contains a cache data RAM with a selectable storage capacity of either 64, 128, 256, or 512 KBytes. The cache data RAM is a direct-mapped memory array, write-through or write-back, that can be organized in either an interleaved or non-interleaved configuration. In addition to the cache data RAM, the L2 cache contains a RAM array that holds the tag address and a dirty bit that is associated with each line of data. Table 17 provides a summary of the L2 cache. A valid bit is not used in this architecture. The L2 cache is programmed via the SCC Register.

Table 17. L2 Cache Features

Feature	Description
Organization	Direct mapped
Capacity	64, 128, 256, or 512 KByte
Data Banks	1 or 2, depending on capacity
Line Size	16 bytes
Tag Size	8 bits
Cacheable Main Memory	8 MBytes to 128 MBytes
Allocation Policy	Allocate on CPU reads; no allocate on writes
Cache Policy	Non-cacheable, write-through (WT), and write-back (WB)

4.12.1 L2 CACHE SIZES/PERFORMANCE

The PSC allows four cache sizes. Table 18 shows the tag and data SRAMs used for various user settings. The PSC supports 15 ns tag SRAMs and 20 ns data SRAMs for the L2 cache at all frequencies. Table 19 shows the range of L2 performance achievable.

Table 18. L2 Options and Component List

Cache Size	Data RAMS	Tag Bits/Cacheable Main Memory	Tag Store
64 KByte	8 8K x 8	A[23:16]/16 MB	4K x 9
128 KByte	4 32K x 8	A[24:17]/32 MB	8K x 9
256 KByte	8 32K x 8	A[25:18]/64 MB	32K x 9
512 KByte	4 128K x 8	A[26:19]/128 MB	32K x 9

Table 19. Performance

Cycle Type	Tag Speed	Data Speed	25 MHz	33 MHz	L1
Interleaved Read	15 ns	15 ns/20 ns	2-1-1-1	2-1-1-1	WT/WB
Non-Interleaved Read	15 ns	15 ns	2-1-1-1	2-2-2-2(1)	WT/WB
Non-Interleaved Read	15 ns	20 ns	2-1-1-1	2-2-2-2	WT/WB
Interleaved Write	15 ns	15 ns/20 ns	2-1-1-1(2)	2-1-1-1(2)	WT/WB
Non-Interleaved Write	15 ns	15 ns/20 ns	2-1-1-1(2)	2-1-1-1(2)	WT
Non-Interleaved Write	15 ns	15 ns/20 ns	3-2-2-2	3-2-2-2	WB

NOTES:
1. 2-1-1-1 may be used only with minimum margin design (light Host Bus loading) and 12 ns Data SRAMs.
2. Programmable option and applies to cache hit dirty write cycles.

4.12.2 CACHE OPERATIONS

During a CPU memory read or write operation, the PSC searches the cache first. Then, if required, it searches main memory for addressed data locations. The L2 cache operation is determined by the cache policy (non-cacheable, write-through, or write-back) as determined by the Secondary Cache Control Register (see Section 3.0, Register Description). If the caching policy is non-cacheable, the cache is not accessed.

Write-through and write-back are two caching policies for updating main memory with data in the cache. For these policies, the cache operation is determined by the type of operation as follows:

CPU Write Cycle

If the caching policy is write-through and there is a cache hit, both the cache and main memory are updated. If there is a cache miss, only main memory is updated. The cache is not updated (no write-allocate).

If the caching policy is write-back and there is a cache hit, only the cache is updated; main memory is not affected. If there is a cache miss, only main memory is updated. The cache is not updated (no write-allocate).

CPU Read Cycle

If there is a cache hit, the cache operation is the same for both write-through and write-back. In this case, data is transferred from the cache to the CPU. Main memory is not accessed.

If there is a cache miss, the line containing the requested data is transferred from main memory to the cache. During the cache line update, a line fill (burst read) memory operation containing four dword transfers occurs on the Host Bus to bring in the new line. This occurs for both write-through and write-back. However, in the case of write-back, if the cache line fill is to a dirty line (D = 1), the dirty line is first written back to main memory before the new line is brought into the cache. For a dirty line write-back operation, the PSC first performs a read from the dirty cache line and writes the data to main memory. Then, the PSC updates the cache (both L1 and L2 simultaneously) with the new line.

4.12.3 CACHE CONSISTENCY

The Snoop mechanism in the PSC ensures data consistency between cache (both L1 and L2 caches) and main memory. Note that, for write-back cache control, the term ''Inquire'' is sometimes used to describe the snooping operation. In this document, the term ''Snoop'' is used for both write-through and write-back cache policies.

The PSC monitors PCI master, ISA master and DMA accesses to main memory and when needed, initiates an inquire (snoop) cycle to the L1 and L2 caches. The snoop mechanism guarantees that consistent data is always delivered to the host CPU, PCI master, ISA master or DMA.

4.12.4 INITIALIZING THE L2 CACHE

The 82420EX PCIset L2 cache architecture does not use a valid bit. Instead, BIOS initializes the L2 cache with valid data. After initialization, the cache controller maintains data coherency between the cache and main memory by keeping all cache lines valid. The PSC cache controller has two special bits to support initialization—Force Hit (SCC Register bit 6) and Force Miss Clean (SCC Register bit 5).

BIOS can use the Force Hit bit to determine the size of the L2 cache. When Force Hit is enabled, BIOS can attempt to alias cache locations on writes. For example, to check a 128 KByte cache size, BIOS writes location ''x'' with value 00h. BIOS can then write location 128k + x with 11h. With Force Hit enabled and the cache in write-back mode, the write does not access main memory. When a value of 00h is read from location ''x'', the L2 cache is greater than 128 KBytes. If 11h is read, the L2 cache is smaller than 128 KBytes. This process is repeated for all cache boundaries.

The Force Miss Clean bit causes all accesses to the L2 cache to be treated as a clean miss. This allows BIOS to initialize the L2. At start-up BIOS enables Force Miss Clean and reads a block of memory equal to the cache size. This initializes the L2 cache with data that is coherent with main memory.

4.12.5 CACHE LINE DESCRIPTION

Each line consists of four dwords of data, a tag field and a Dirty (D) bit. The tag field and control bits are read/written by the PSC during normal cache operations and are not accessible by software.

D: Dirty

The Dirty bit is set to 1 by the PSC to indicate that data modified in the cache line has not been written back to main memory.

Tag: Real Address Tag

The PSC uses the Tag field for cache line hit/miss determination. The width of the Tag field is fixed at 8 bits. The table below shows the real address bits that are stored in the Tag field as a function of the cache sizes.

Cache Size	8-Bit Tag
64 KBytes	A[23:16]
128 KBytes	A[24:17]
256 KBytes	A[25:18]
512 KBytes	A[26:19]

Doubleword[3:0]

Each line of the cache data RAM contains four dwords.

4.12.6 L2 CACHE STRUCTURE

The tag is 8 bits plus a dirty bit. Either interleaved or non-interleaved organizations are permitted. The PSC will assert the COE[1:0]# and CWE[1:0]# signals to both banks, even when programmed for non-interleaved mode. The interleaved L2 can provide zero wait-state reads and writes. Figure 12 shows the interconnection between the PSC and an interleaved L2. The PSC has a variety of programmable access timings to support 25 MHz and 33 MHz. These options are controlled by the Secondary Cache Control Register (SCC).

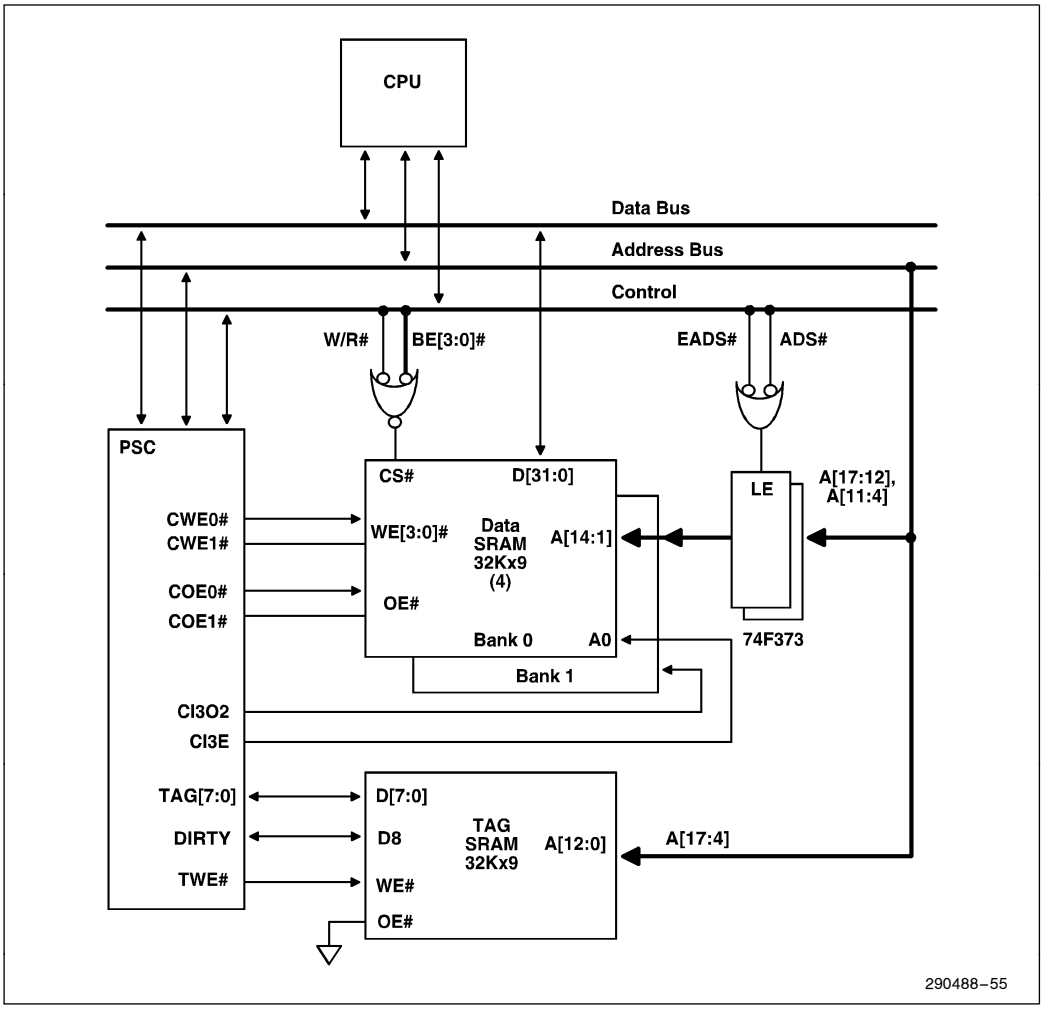


Figure 12. The PSC with an Interleaved L2 (256 KBytes Data)

4.13 Dram Interface

The DRAM controller interfaces main memory to the Host Bus, PCI Bus, and ISA Bus. The PSC provides the control signals, address lines, and data path control. A two-way interleaved DRAM organization is supported for optimum main memory performance. Up to ten single-sided SIMMs or four double-sided and two single-sided SIMMs provide a maximum of 128 MBytes of main memory. The DRAM controller interface is fully configurable through a set of control registers (the DRAM Control Mode Register, the DRAM Memory Hole Register, and the five DRAM Row Boundary [DRB] Registers).

The PSC controls a 64-bit memory array (72-bit including parity) and/or a 32-bit memory array (36-bit including parity) ranging in size from 1 to 128 MBytes using industry standard 36-bit wide memory modules with fast page-mode DRAMs. Both single- and double-sided SIMMs are supported. The eleven multiplexed address lines (MA[10:0]) permit the use of 256Kx36, 1Mx36, and 4Mx36 SIMMs. Both interleaved and non-interleaved rows are supported simultaneously. Five RAS# lines enable up to five rows of DRAM. Eight CAS# lines allow byte control over the array during read and write operations. The PSC supports 70 ns DRAMs. Page mode accesses efficiently transfer data in bursts. Parity support is optional.

The PSC DRAM performance is controlled through programmable wait-states. Various DRAM timing parameters may be set in the DRAM Control Register. Programmable timings support 70 ns DRAMs at 25 MHz and 33 MHz. Programmable parameters include RAS precharge, CAS precharge, CAS low time, MA setup time, and MA hold time. The PSC provides RAS only refresh, de-coupled from ISA refresh, and hidden from any access.

4.13.1 DRAM ADDRESS TRANSLATION

The multiplexed row/column address to the DRAM memory array is provided by the MA[10:0] signals and is derived from the host address bus as defined by Table 20. The page size is 2 KBytes for non-interleaved rows and 4 KBytes for interleaved rows. The page offset address is driven over the MA[8:0] lines when driving the column address. In non-interleaved rows the PSC drives address bit 2 on the MA8 line, minimizing the multiplexing required. The MA[10:0] lines are translated from the address lines A[24:3] for all memory accesses.

4.13.2 DRAM STRUCTURE

Figure 13 illustrates an 8-SIMM configuration supporting single-sided SIMMs. A row in the DRAM array is made up of two SIMMs that share a common RAS# line. SIMM0 and SIMM1 comprise row 0 and are connected to RAS0#. Within any given row, the two SIMMs must be the same size. Among the four rows, SIMM densities can be mixed in any order (i.e., there are no restrictions on the ordering of SIMM densities among the four rows). Any row may also contain a single SIMM (non-interleaved). This allows the user to upgrade the 82420EX PCIset platform one SIMM at a time. Each row is controlled by up to 8 CAS lines. Any row that is populated with only one SIMM must be connected to the low order CAS lines (CAS[3:0]#). The MA[10:0] and WE# lines should be externally buffered if a load of more than two double-sided SIMMs is implemented. Two buffered copies of the signals are recommended to drive the four row array. Three buffered copies of the signals are recommended to drive the five row array.

Figure 14 illustrates a 3-SIMM configuration using one single-sided SIMM in row one, and two double sided SIMMs in row 2. In this configuration, single- and double-sided SIMMs can be mixed. For example, if a single-sided SIMM is installed into the socket marked SIMM0 (connected to RAS0#) and RAS1# is not connected, row 0 is then populated and row 1 is empty. Two double-sided SIMMs could then be installed in the sockets marked SIMM2 and SIMM3, populating rows 2 and 3. For systems with no more than 2 SIMMs, the 244's buffering MA[10:0] and WE#, as well as the 245's on the host data bus, may be omitted. (Note that the 245's on the Host Data Bus are recommended at 50 MHz, regardless of the number of SIMMs.)

Table 20. DRAM Address Translation for Interleaved Rows

MA[10:0]	10	9	8	7	6	5	4	3	2	1	0
Interleaved Rows											
Column Address	A23	A21	A11	A10	A9	A8	A7	A6	A5	A4	A3
Row Address	A24	A22	A20	A19	A18	A17	A16	A15	A14	A13	A12
Non-Interleaved Rows											
Column Address	A23	A21	A2	A10	A9	A8	A7	A6	A5	A4	A3
Row Address	A22	A20	A11	A19	A18	A17	A16	A15	A14	A13	A12

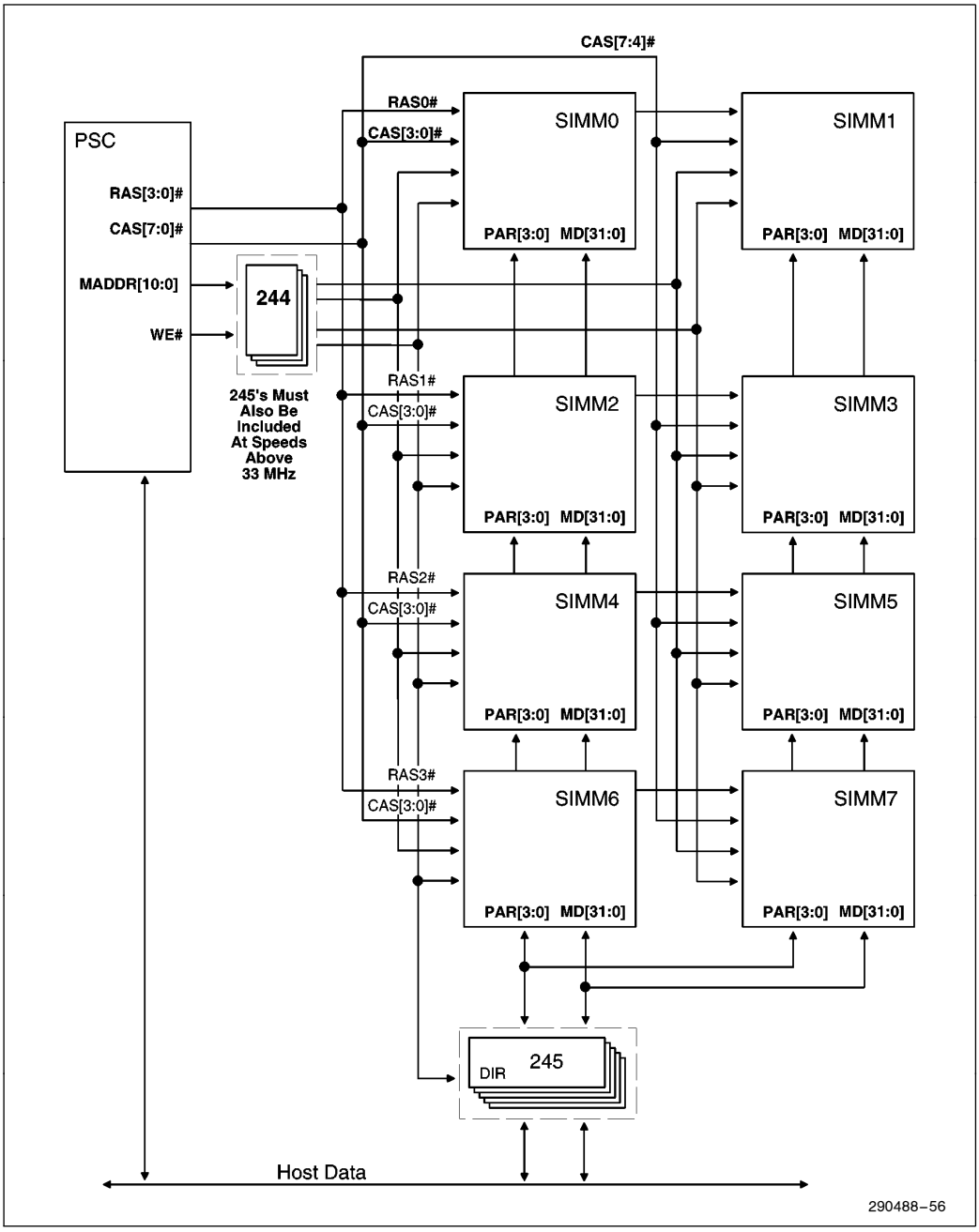


Figure 13. 8-SIMM Configuration Supporting Single-Sided SIMMs

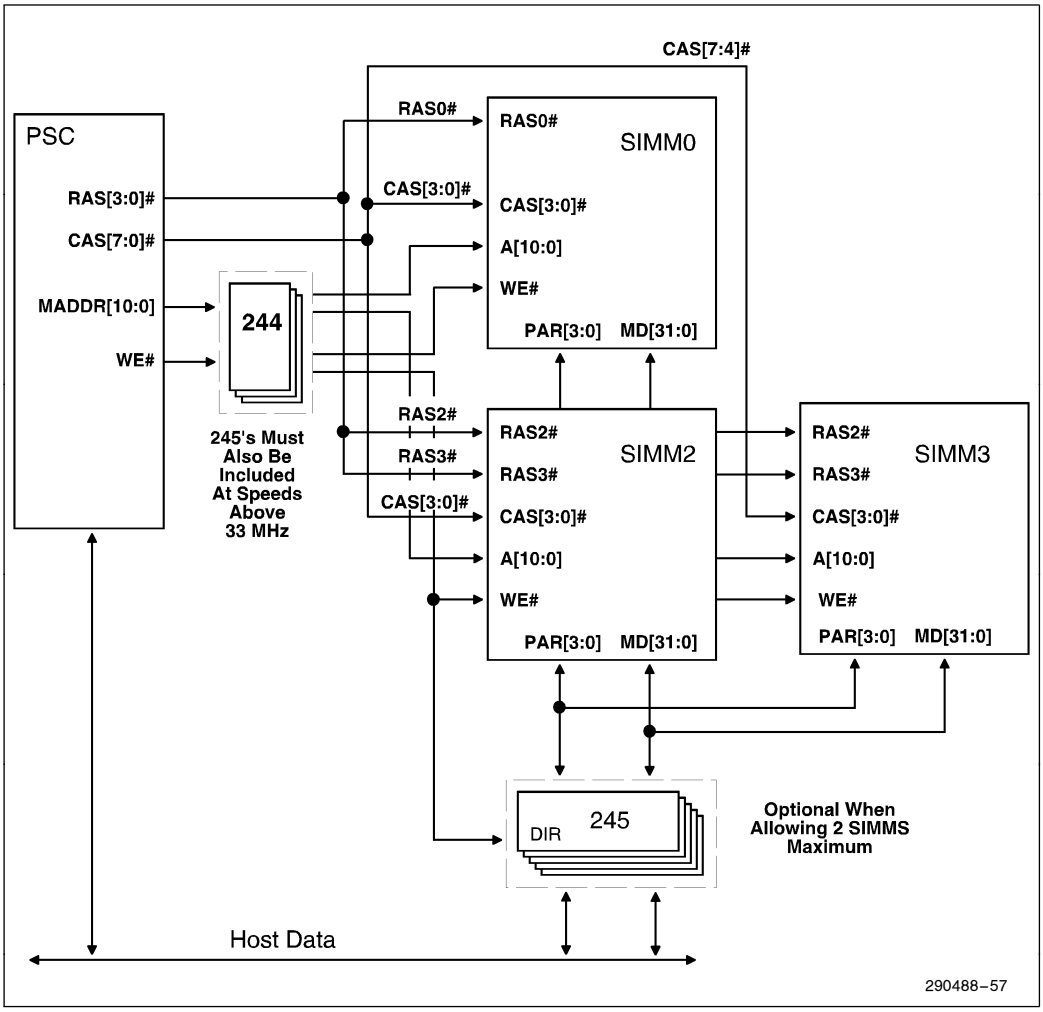


Figure 14. 3-SIMM Configuration (One Single-Sided SIMM in Row One and Two Double-Sided SIMMs in Row Two)

4.13.3 DRAM SIMM SIZE/DENSITY OPTIONS

Providing support for both interleaved and non-interleaved rows gives the user a wide range of DRAM population options. In any row the PSC supports address depths of 256K, 1M, or 4M. Each row may be populated with one or two SIMMs.

4.13.4 DRAM PAGE MODE

The PSC may be programmed to leave the RAS lines active after a DRAM access for faster page mode accesses. The mode is programmed in the DRAM Control Register. When Page Mode is enabled, the RAS lines remain active after the access.

The next access is considered a page hit if the access is to the same page. The PSC has a 2 KByte page size for non-interleaved rows and a 4 KByte page size when accessing an interleaved row. If the page mode is not enabled all accesses are row misses.

4.13.5 PROGRAMMABLE WAIT-STATES

4.13.5.1 RAS Precharge

RAS precharge impacts page miss accesses, as well as the refresh time. In a page miss, the active RAS line must be negated and a new row address strobed into the DRAMs. When negated, RAS precharge determines the number of cycles before the RAS line can be re-asserted. Similarly, the RAS precharge determines the time RAS must be negated before and after refresh.

4.13.5.2 CAS Read Time

CAS read time indicates how long to leave CAS low after asserted during a DRAM read and how long it is negated between access. In the case of interleaved DRAM access, the CAS high time implied by this setting is ignored as this high time is more constrained by the opposite banks low time.

4.13.5.3 CAS Write Time

CAS write time controls the CAS waveform for DRAM writes. The high time defined by the CAS write time setting is ignored for interleaved rows.

4.13.5.4 MA Setup Time

The MA setup time defines the number of cycles after MA is switched before RAS or CAS are driven active. DRAMs latch row and column address when RAS and CAS fall. The setup time of the addresses to RAS/CAS for all DRAMs is 0 ns. The PSC supports direct drive of the MA lines, which removes any external logic between the MA on the PSC and the DRAM. When there is no external buffering, the MA-to-DRAM path and the CAS-to-DRAM path are well matched. In these cases an aggressive MA setup time can be programmed. When the external buffers are present, there could be mismatch in the paths, and a more conservative MA setup should be chosen.

4.13.5.5 MA Hold Time

The MA hold time defines the number of clocks after RAS or CAS have been asserted before MA can be changed. This value is determined in a manner similar to MA setup time. DRAM requirements for 60 ns and 70 ns DRAMs range from 10 ns to 15 ns.

4.13.6 DRAM PERFORMANCE

Table 21 summarizes DRAM performance for various programming options for both 60 ns and 70 ns DRAMs. Other cycle constraints that are met by design include DRAM access from RAS falling and DRAM access from row address, and many others. All accesses shown below assume no wait-states required for other Host Bus devices (L2, etc.). If, as discussed in the previous section, buffers must be turned around and contention with other host devices avoided, the minimum lead off for read page hits will be lengthened by one cycle.

Table 21. DRAM Performance

System	RAS pre-charge	CAS Read	CAS Write	MA Setup	MA Hold	Clock Freq.	Performance No L2	Performance With L2
60 ns DRAM Min	1	1/1	1/1	.5	.5	25 MHz	Ird: 3/5/5-1-1-1 NIrd: 3/5/5-2-2-2 wr: 3/5/5-2-2-2	Ird: 3/5/5-1-1-1 NIrd: 3/5/5-2-2-2 Same as No L2
Margin	1.5	1.5/.5	1/1	1	.5	33 MHz	Ird: 3/6/7-2-2-2 NIrd: 3/6/7-2-2-2 wr: 3/5/6-2-2-2	Ird: 4/6/7-2-2-2 NIrd: 4/6/7-2-2-2 Same as No L2
60 ns DRAM High	1.5	1.5/.5	1/1	.5	.5	25 MHz	Ird: 3/5/6-2-2-2 NIrd: 3/5/6-2-2-2 wr: 3/5/6-2-2-2	Ird: 4/5/6-2-2-2 NIrd: 4/5/6-2-2-2 Same as No L2
Margin	1.5	1.5/.5	1/1	1	1	33 MHz	Ird: 3/7/7-2-2-2 NIrd: 3/7/7-2-2-2 wr: 3/6/7-2-2-2	Ird: 4/7/7-2-2-2 NIrd: 4/7/7-2-2-2 Same as No L2
70 ns DRAM Min	1.5	1.5/.5	1/1	.5	.5	25 MHz	Ird: 3/5/6-2-2-2 NIrd: 3/5/6-2-2-2 wr: 3/5/6-2-2-2	Ird: 4/5/6-2-2-2 NIrd: 4/5/6-2-2-2 Same as No L2
Margin	2	1.5/.5	1/1	1	.5	33 MHz	Ird: 3/6/7-2-2-2 NIrd: 3/6/7-2-2-2 wr: 3/5/7-2-2-2	Ird: 4/6/7-2-2-2 NIrd: 4/6/7-2-2-2 Same as No L2
70 ns DRAM High	2	1.5/.5	1/1	.5	.5	25 MHz	Ird: 3/5/7-2-2-2 NIrd: 3/5/7-2-2-2 wr: 3/5/6-2-2-2	Ird: 4/5/7-2-2-2 NIrd: 4/5/7-2-2-2 Same as No L2
Margin	3	2/1	1/1	1	.5	33 MHz	Ird: 4/7/9-2-2-2 NIrd: 4/7/9-3-3-3 wr: 3/6/8-2-2-2	Ird: 4/7/9-2-2-2 NIrd: 4/7/9-3-3-3 Same as No L2

NOTES
I = Interleaved
NI = Non-Interleaved

4.14 Power Management

The 82420EX PCIset has extensive power management capability permitting a system to operate in a low power state without being powered down. In a typical desktop personal computer there are two states—Power On and Power Off. Leaving a system powered on when not in use wastes power. The 82420EX PCIset provides a Fast On/Off feature that creates a third state called Fast Off (Figure 15). When in the Fast Off state, the system consumes less power than the Power On state.

The 82420EX PCIset's power management architecture is based on three functions—System Management Mode (SMM), Clock Control, and Advanced Power Management (APM). Software (called SMM code) controls the transitions between the Power On state and the Fast Off state. The IB invokes this software by generating an SMI to the CPU (asserting the SMI# signal). A variety of programmable events are provided that can generate an SMI. The SMM code places the system in either the Power On state or the Fast Off state.

A Fast On event is an event that instructs the computer (via an SMI to the CPU) to enter the Power On state in anticipation of system activity by the user. Fast On events are programmable and include moving the mouse, pressing a key on the keyboard, an external hardware event, an incoming call to a system FAX/Modem, a RTC alarm, or the operating system.

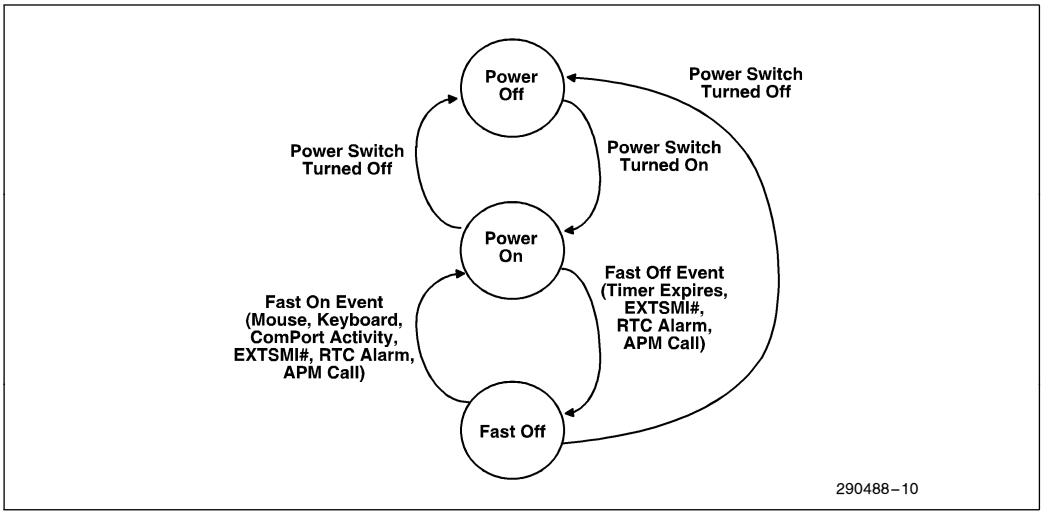


Figure 15. Fast On/Off Flow

4.14.1 SMM MODE

SMM mode is invoked by asserting the SMI# signal to the CPU. The PCIset provides a variety of programmable events that can generate an SMI. When the CPU receives an SMI, it enters SMM mode and executes SMM code out of SMRAM. The SMM code places the system in either the Power On state or the Fast Off state. In the Power On state, the computer system operates normally. In this state one of the four programmable events listed below can trigger an SMI.

1. A global idle timer called the Fast Off timer expires (an indication that the end user has not used the computer for a long period of time).
2. The EXTSMI# pin is asserted.
3. A RTC alarm interrupt is detected.
4. The operating system issues an APM call.

4.14.2 SMI SOURCES

The SMI# signal can be asserted by hardware events, an external SMI event (EXTSMI#), and software events (via the APMC and APMS Registers). Enable/disable bits (in the SMIEN Register) permit each event to be individually masked from generating an SMI. In addition, the SMI# signal can be globally enabled/disabled in the SMICNTL Register. Status of the individual events causing an SMI is provided in the SMIREQ Register. For detailed information on the SMI control/status registers, refer to Section 3.0, Register Description.

Hardware Interrupt Events

Hardware events are enabled/disabled from generating an SMI in the SMIEN Register. The hardware events consist of IRQ[12,8#,4,3,1] and the Fast Off Timer. When enabled, the occurrence of the corresponding hardware event generates an SMI (asserts the SMI# signal), regardless of the current power state of the system.

Fast Off Timer

The Fast Off Timer is used to indicate (through an SMI) that the system has been idle for a pre-programmed period of time. The timer counts down at a selectable rate from a programmed start value and when the count reaches 00h, an SMI is generated. The timer decrement rate can be set to 1 count every minute, ms, or HCLKIN (via the SMICNTL Register) and is re-loaded each time a System Event occurs.

System events are programmable events that can keep the system in the Power On state when there is system activity. These events are indicated by the assertion of IRQ[15:9,8#,7:3,1:0], NMI, or SMI signals.

In addition to system events, *break events* cause the system to transition from a Fast Off state to the Power On state. System events (and break events) are enabled/disabled in the SEE Register. When enabled and the associated hardware event occurs (signal is asserted), the Fast Off Timer is re-loaded with its initial count.

EXTSMI#

The EXTSMI# input pin provides the system designer the capability to invoke SMM with external hardware. For example, the EXTSMI# input could be connect to a ''green button'' permitting the user to enter the Fast Off state by depressing a button. The EXTSMI# generation of an SMI is enabled/disabled in the SMIEN Register.

Software Events

Software events (accessing the APMx Registers) indicate that the OS is passing power management information to the SMI handler. There are two Advanced Power Management (APM) registersAPM Control (APMC) and APM Status (APMS) Registers. These registers permit software to generate an SMI; by writing to the APMC Register. For example, the APMC can be used to pass an APM command between APM OS and BIOS and the APMS Register could be used to pass data between the OS and the SMI handler. For detailed descriptions of these registers See Section 3.0, Register Description. Note that the two APM Registers are located in normal I/O space. The remaining power management registers are located in PCI configuration space.

4.14.3 SMI (SMI#) AND INTERRUPT (INTR) ORDERING

To maintain the SMI#/INTR order, an interrupt blocking mechanism has been incorporated into the IB. The blocking mechanism blocks interrupt requests that can generate an SMI# from being processed by the interrupt controller until the SMI# has been serviced by the SMM code. This blocking mechanism is selective and only affects the IRQ[12,8#,4,3,1] signals that are enabled to generate an SMI# (via the SMIEN Register). In addition, the blocking mechanism is only invoked if the SMI# signal is unmasked (via the SMICNTL Register). Note that PIRQ[1,0]s routed to one of the dual-purpose interrupt request lines are also affected by the blocking mechanism. Thus, the following criteria applies to the blocking mechanism:

1. The assertion of IRQ[12,8#,4,3,1] are blocked if the interrupt request line is programmed for SMI (i.e., the interrupt request line is enabled for SMI via the SMIEN Register and the SMI# signal is not masked via the CSMIGATE bit in the SMICNTL Register).
2. A blocked IRQ request is unblocked and processed by the interrupt controller when software masks the SMI# signal by setting the CSMIGATE bit to 0 in the SMICNTL Register.
3. IRQs that are already asserted when SMI# is unmasked (CSMIGATE set from a 0 to 1) are not blocked and are processed by the interrupt controller.

The following are BIOS and hardware considerations regarding the SMI#/INTR ordering:

- To process blocked, active IRQs, software (SMM code only) should mask the SMI# signal. If SMI# is masked outside SMM code when an IRQ that can generate an SMI# and the INTR signal is active, the SMI# and INTR order is not guaranteed.
- The SMI software handler should use the SMIREQ Register status bits and not the interrupt controller IRR to dispatch the routine (vector to the appropriate SMI function). By using the SMIREQ Register, the SMI handler has the freedom to mask the SMI# signal before or after the execution of the SMI function. Note that the IRR is updated only when the SMI# signal is masked.
- The IB updates new active SMI sources while the system is in SMM, independent of the state of the mask/unmask of the SMI# signal. When the SMI handler completes the execution of a certain SMI function, it should check whether other active SMI sources exist and service these sources before executing the **RSM** instruction.
- When the SMIREQ Register indicates that all SMI sources are inactive, the SMI handler should unmask the SMI# signal and execute the **RSM** instruction. Note that, all active SMI sources (status bits not set to 0 in the SMIREQ Register), will generate a new SMI# to the CPU when SMI# is unmasked.
- The SMI handler should not check for active SMI sources, or execute the new sources, after the SMI# signal is unmasked. Such an SMI source will generate a new active SMI# and the CPU will latch the new SMI# (and recognize it after **RSM**). Thus, executing the SMI source before **RSM** will cause a spurious SMI# after the **RSM** execution.

4.14.4 CLOCK CONTROL

The CPU can be put in a low power state by asserting the STPCLK# signal. STPCLK# is an interrupt to the CPU. However, for this type of interrupt, the CPU does not generate an interrupt acknowledge cycle. Once the STPCLK# interrupt is executed, the CPU enters the Stop Grant state. In this state, the CPU's internal clocks are disabled and instruction execution is stopped. The Stop Grant state is exited when the STPCLK# signal is negated.

Software can assert STPCLK#, if enabled via the SMICNTL Register, by a read of the APMC Register. Note that STPCLK# can also be periodically asserted by using clock throttling as described below.

The IB automatically negates STPCLK# when a break event occurs (if enabled in the SEE Register) and the CPU stop grant special cycle has been received. Software can negate STPCLK# by disabling STPCLK# in the SMICNTL Register or by a write to the APMC Register.

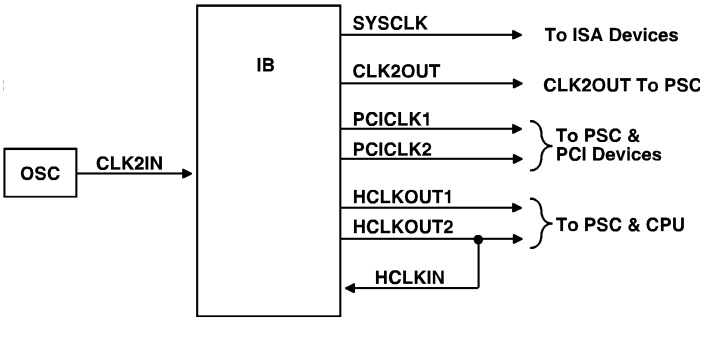
Clock Throttling (Emulating Clock Division)

Clock throttling permits the IB to periodically place the CPU in a low power state. This emulates clock division. When clock throttling is enabled, the CPU runs at full frequency for a pre-defined time period and then is stopped for a pre-defined time period. The Run/Stop time interval ratio emulates the clock division effect from a power/performance point of view. However, clock throttling is more effective than dividing the CPU frequency. For example, if the CPU is in the Stop Grant state and a system break event occurs, the CPU clock returns to full frequency. In addition, there is no recovery time latency to start the clock.

Two programmable 8-bit clock throttle timer control registers set the STPCLK# high (negate) and low (assert) times—the CTLTMRH and CTLTMRL Registers. The timer is clocked by a 32 μs internal clock. This allows a programmable timer interval for both the STPCLK# high and low times of 0-8 ms.

4.15 Clocks

The IB contains a clock generation unit that generates the system clocks. The IB generates HCLKOUTx (host clocks), PCICLKx (PCI clocks), SYSCLK (ISA System clock), and CLK2OUT (delayed version of CLK2IN) signals to the system (Figure 16). An external clock driver is not required. Two HCLKOUT signals and two PCICLK signals are provided to drive the loads. One of the HCLKOUT outputs is fed back to the HCLKIN pin to become the IB clock. CLK2OUT is used by the PSC. The IB uses a 2X clock input (CLK2IN) as the source to generate the system clocks. For CPU or PCI initiated cycles to the ISA Bus, SYSCLK is stretched to synchronize the falling edge BALE to the rising edge of the SYSCLK.



NOTE:
HCLKIN can be connected to either HCLKOUT1 or HCLKOUT2.

Figure 16. System Clock Distribution

There are three clock configurations (strapping options) that set the clock divisors as shown in Table 22. The IB samples the CMDV# and SIDLE# signals on the rising edge of PWROK to determine the clock divisor value. The IB CLK2IN pin is divided by either 1 or 2 to generate HCLKOUT[2,1] and PCICLK[2,1]. One of the HCLKOUT signals is fed back to the HCLKIN input of the IB and divided by either 3, 4, 5, or 6, to generate SYSCLK. Note that the configuration information provided in Table 22 is software accessible in the Host Bus Select Register.

4.15.1 CLOCK LAYOUT/LOADING RECOMMENDATION

A spice analysis was done on the HCLKOUT, PCICLK, and CLK2OUT signals to determine the loading requirements necessary to maintain the clock rising edge skew values shown below. The spice analysis was done at 100°C with a V_{CC} of 4.75V. These are worst case conditions considering the rising clock edge skew is the most critical. The motherboard impedance was varied from 50Ω to 90Ω.

Rising Edge Clock Skew Results/Recommendation

Clocks	Max. Skew Seen at the Receiver	Measured at
HCLK to HCLK	1.0 ns	1.5V to 1.5V(5)
CPU HCLK to PSC HCLKIN	1.0 ns	1.5V to 1.5V(5)
PCICLK to PCICLK	2.0 ns	1.5V to 1.5V(5)
PSC PCICLKIN to PSC HCLKIN	0.5 ns	1.5V to 2.5V(5)
HCLKOUT to PCICLK	2.0 ns	1.5V to 1.5V(5)

NOTES:
1. The skew values in the above table include an IB intrinsic skew between clock outputs of 0.5 ns.
2. To achieve the 0.5 ns clock skew shown for the CPU HCLKOUT to PSC HCLKOUT, it is required that the CPU's HCLKOUT and PSC's HCLKOUT use the same HCLKOUT from the IB.
3. The above skews were determined using a CPU load of 5 pF–15 pF. If an Intel486 SX is used, add 0.5 ns to the skew measurements for HCLKOUT to HCLKOUT and CPU HCLKOUT to PSC HCLKOUT. This is necessary because the clock capacitive loading for the Intel486 SX varies from 5 pF–20 pF.
4. Assuming that the series resistors are equal and loads are equal, the skew will vary 0.2 ns for every 1″ difference in trace length seen between the clocks. Assuming that the series resistors are equal and the trace lengths are equal, the skew will vary 0.2 ns for every 5 pF difference in loading seen between the clocks. The skew is measured at the receiver.
5. All clock skew measurements were made from the 1.5V to 1.5V level on the rising edge of the clocks, with the exception of the PCICLK and HCLK inputs to the PSC. The skew between the HCLK input to the PSC and the PCICLK input to the PSC is measured from 1.5V on the rising edge of PCICLKIN to 2.5V on the rising edge of HCLKIN, as measured at the PSC. This skew must not exceed 0.5 ns.

Table 22. Clock Configurations

Strapping Options		CLK2IN	HCLKOUT	HCLKOUT Divisor	PCICLK	PCICLK Divisor	SYSCLK	SYSCLK Divisor
SIDLE#	CMDV#							
0	0	50 MHz	25 MHz	2	25 MHz	2	8.33 MHz	3
0	1	66 MHz	33 MHz	2	33 MHz	2	8.25 MHz	4
1	0	Reserved						
1	1	Reserved						

RECOMMENDED OPTIONS

In the following recommended option the series resisters shown in each option are required to improve the relative skew between clocks, and to limit the amount of ringing and undershoot seen on the lines. The undershoot was limited to approximately 0.5V to 0.8V, worst case voltage and temperature.

All the traces are labeled with a possible trace length ±0.3″ and a relative comparison of lengths using ''A'' ±0.3″ as shown in Figure 17 where ''A'' equals a route length of about 7.8″ (chosen by the PCB layout engineer). ''A'' should be kept between 5″ and 10″.

5.0 DESIGN CONSIDERATIONS

Design considerations are chip set related issues which affect all 82420EX PCIset designs. See your Intel representative and the 82420EX PCIset Value Flexible Motherboard Design Guide (297460) for the latest version of the design considerations.

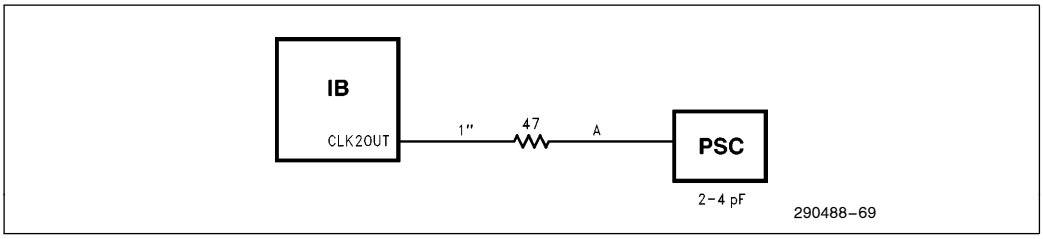


Figure 17. CLK2OUT OPTION

The following option includes two HCLKOUT loads + the IB.

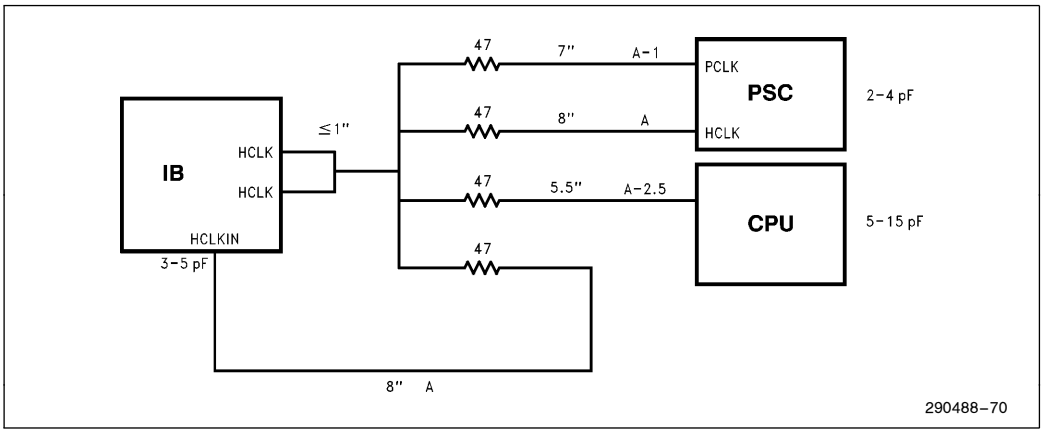


Figure 18. HCLKOUT OPTION

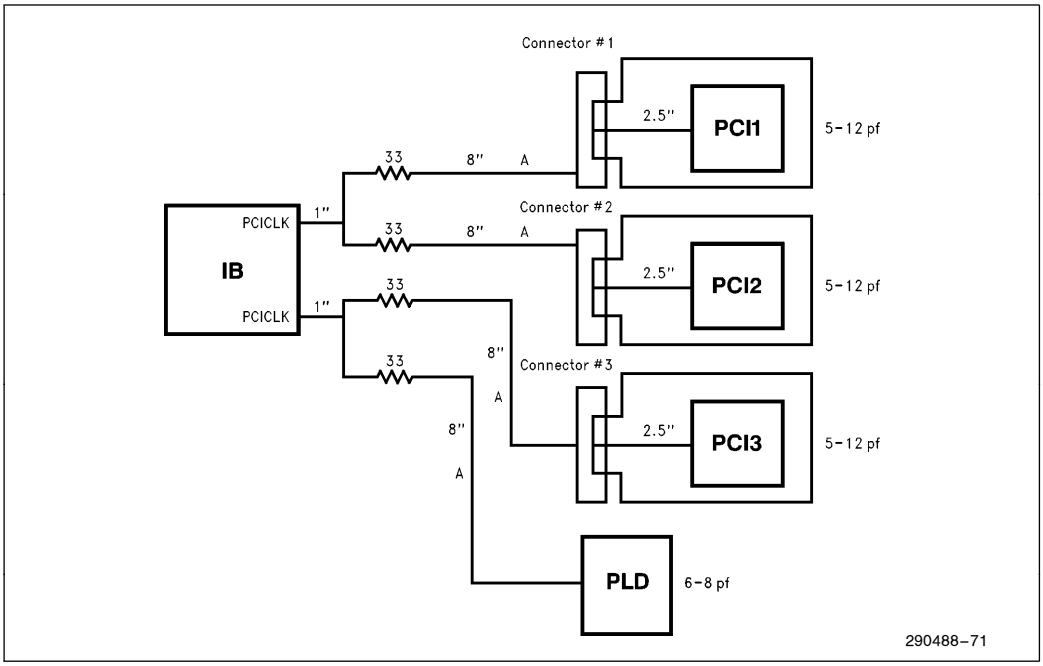


Figure 19. PCICLK OPTION

6.0 ELECTRICAL CHARACTERISTICS

This section provides the 82420EX PCIset maximum ratings, DC characteristics and AC characteristics including timing diagrams.

NOTICE: This data sheet contains information on products in the sampling and initial production phases of development. The specifications are subject to change without notice. Verify with your local Intel Sales office that you have the latest data sheet before finalizing a design.

**WARNING: Stressing the device beyond the "Absolute Maximum Ratings" may cause permanent damage. These are stress ratings only. Operation beyond the "Operating Conditions" is not recommended and extended exposure beyond the "Operating Conditions" may affect device reliability.*

6.1 Maximum Ratings

Case Temperature Under Bias . . . −65°C to +110°C

Storage Temperature . . . . . −65°C to +150°C

Supply Voltages
with Respect to Ground . . . −0.5V to V_{CC} + 0.5V

Voltage on Any Pin . . . . . −0.5V to V_{CC} + 0.5V

Power Consumption (IB) 0.5W

Power Consumption (PSC) 1.0W

6.2 PSC and IB DC Characteristics

DC Characteristics (V_{DD} = 5V ± 5%, T_{CASE} = 0 to 85°C)

Symbol	Parameter	Min	Max	Units	Test Conditions	Notes
V_{IL1}	Input Low Voltage	−0.5	0.8	V		
V_{IH1}	Input High Voltage	2.0		V		
V_{IL2}	Input Low Voltage		0.3*V_{CC}	V		10
V_{IH2}	Input High Voltage	0.7*V_{CC}		V		10
V_{T+}	TTL Schmitt Trigger, Rising Threshhold	1.9		V	V_{CC} = 5 V	11
V_{T-}	TTL Schmitt Trigger, Falling Threshhold		1.3	V	V_{CC} = 5 V	11
V_{T+}	Hysteresis Voltage	600		mV	V_{CC} = 5 V	11
V_{OL1}	Output Low Voltage (IB)		0.45	V	I_{OL} = 24 mA	1
V_{OH1}	Output High Voltage (IB)	2.4		V	I_{OH} = −3.0 mA	1
V_{OL2}	Output Low Voltage (IB)		0.4	V	I_{OL} = 4 mA	2
V_{OH2}	Output High Voltage (IB)	2.4		V	I_{OH} = −2 mA	2
V_{OL3}	Output Low Voltage (IB)		0.4	V	I_{OL} = 8 mA	3
V_{OH3}	Output High Voltage (IB)	2.4		V	I_{OH} = −2 mA	3
V_{OL4}	Output Low Voltage (PSC)		0.45	V	I_{OL} = 8 mA	4
V_{OH4}	Output High Voltage (PSC)	2.4		V	I_{OH} = −2 mA	4
V_{OL5}	Output Low Voltage (PSC)		0.45	V	I_{OL} = 4 mA	5
V_{OH5}	Output High Voltage (PSC)	2.4		V	I_{OH} = −2.0 mA	5
V_{OL6}	Output Low Voltage (PSC)		0.4	V	I_{OL} = 4 mA	6
V_{OH6}	Output High Voltage (PSC)	2.4		V	I_{OH} = −4 mA	6
V_{OL7}	Output Low Voltage		0.4	V	I_{OL} = 6 mA	7
V_{OH7}	Output High Voltage	2.4		V	I_{OH} = −2.0 mA	7
V_{OL8}	Output Low Voltage		0.4	V	I_{OL} = 4 mA	8
V_{OH8}	Output High Voltage	0.9*V_{CC}		V	I_{OH} = −2 mA	8
V_{OL9}	Output Low Voltage		0.4	V	I_{OL} = 8 mA	9
V_{OH9}	Output High Voltage	0.9*V_{CC}		V	I_{OH} = −2mA	9
I_{LI1}	Input Leakage Current		±10	A	0V < V_{IN} < V_{CC}	
I_{LI2}	Input Leakage Current		−350	A	0V < V_{IN} < V_{CC}	12
I_{LO}	Output Leakage		±10	A	0.45 < V_{IN} < V_{CC}	

DC Characteristics (V_{DD} = 5V ± 5%, T_{CASE} = 0 to 85°C) (Continued)

Symbol	Parameter	Min	Max	Units	Test Conditions	Notes
C_{IN}	Capacitance Input		9	pF	@ 1 MHz	
C_{OUT}	Capacitance Output or I/O		9	pF	@ 1 MHz	
I_{CC}	V_{CC} Supply Current (IB)		100	mA		
I_{CC}	V_{CC} Supply Current (PSC)		200	mA		

NOTES:
1. V_{OL1}, V_{OH1} = SD[15:0], SA[19:0], LA[23:17], SBHE#, MEMR#, MEMW#, AEN, SPKR, BALE, SYSCLK, IOR#, IOW#, SMEMR#, SMEMW#, RSTDRV, REFRESH#, IOCHRDY, MEMCS16#, TC, DACK#
2. V_{OL2}, V_{OH2} = XBUSTR#, XBUSOE#, IGNNE#, RTCCS#, KBCCS#, BIOSCS#, RTCALE, INTR, NMI, CMDV#, SIDLE#, LREQ#, SMI#, STPCLK#
3. V_{OL3}, V_{OH3} = A[17:2], HCLKOUT1, HCLKOUT2, PCICLK1, PCICLK2, CPURST, PCIRST#
4. V_{OL4}, V_{OH4} = A[31,26:2], HD[31:0], HDP[3:0], BE[3:0]#, W/R#, RDY#, BRDY#, CI3E, CI3O2, CWE[1:0]#, COE[1:0]#, MA[10:0], RAS[4:0]#, CAS[7:0]#, WE#, LBIDE#
5. V_{OL5} V_{OH5} = HOLD, AHOLD, EADS#, KEN#, TWE#, TAG[8:0], AD[31:0], C/BE[3:0]#, PAR, CMDV#, SIDLE#, LGNT#, PGNT[1:0]
6. V_{OL6}, V_{OH6} = SRESET/INIT
7. V_{OL7}, V_{OH7} = FRAME#, IRDY#, TRDY#, STOP#, SERR#, DEVSEL#
8. V_{OL8}, V_{OH8} = CMDV#, SIDLE#, LREQ#, LGNT#
9. V_{OL9}, V_{OH9} = CLK2OUT, PCICLK1, PCICLK2, HCLKOUT1, HCLKOUT2
10. V_{IL2}, V_{IH2} = HCLKIN (IB), CMDV#, SIDLE#, LGNT#
11. This applies to PWROK, EXTSMI#, FERR#, IRQ[1,3-7,9-11,14,15],IRQ12M, PIRQ[1:0].
12. This applies to pins that include a weak internal pull-up (IRQ8#[IB], FERR#[IB], D/C#[PSC], ADS#[PSC], BLAST#[PSC], HITM#[PSC], SMI#[PSC], SMIACT#[PSC]).

6.3 IB AC Characteristics

This section provides the AC parameters and timing diagrams.

6.3.1 CLOCK/RESET TIMINGS

AC Characteristics (V_{DD} = 5V ± 5%, T_{CASE} = 0 to 85°C)

Symbol	Parameter	Min	Max	Units	Notes	Fig
t1a	HCLKIN Period Stability		0.1%			
t1b	HCLKIN Period	30.0	40.0	ns		18
t1c	HCLKIN High/Low Time	8.0		ns		18
t1d	HCLKIN Rise/Fall Time		2.0	ns		18
t1e	HCLKOUT[2,1] Period	30.0	40.0	ns		18
t1f	HCLKOUT[2,1] High/Low Time	12.0		ns		18
t1g	HCLKOUT[2,1] Rise/Fall Time		2.0	ns		18
t1h	CLK2IN Period Stability		0.1%			
t1i	CLK2IN Period	15	20.0	ns		18
t1j	CLK2IN High/Low Time	4.0		ns		18
t1k	CLK2IN Rise/Fall Time		1.5	ns		18
t1l	CLK2OUT Period	15	20	ns		18
t1o	PCICLK[2,1] Period	30	40	ns		18
t1p	PCICLK[2,1] High/Low Time	12.0		ns		18
t1q	PCICLK[2,1] Rise/Fall Time		3.0	ns		18
t1r	OSC Period	67	70	ns		18
t1s	OSC High/Low Time	20		ns		18
	ISA CLOCK TIMINGS					
	SYSCLK					
t1t	Period	120	125	ns		18
t1u	High/Low time	56		ns		18
t1v	Rise/Fall time		4	ns		18
	MISCELLANEOUS CLOCK TIMINGS					
t1w	PCICLK to HCLKOUT Skew		0.4	ns	1	19
	RESET TIMINGS					
t1x	CPURST, PCIRST#, RSTDRV Driven Inactive After PWROK is Driven Active High.	2		HCLKIN		20
t1y	CPURST, PCIRST#, RSTDRV Active Pulse Width. Initiated via the TRC Register.	1		ms		21
t1z	CPURST Valid Delay from HCLKIN Rising	3	17	ns		

NOTES:
1. Except when STPCLK# is active.

6.3.2 PSC/IB LINK INTERFACE TIMINGS

AC Characteristics (V_{DD} = 5V ± 5%, T_{CASE} = 0 to 85°C)

Symbol	Parameter	Min	Max	Units	Notes	Fig
t2a	CMDV#, SIDLE# Setup to HCLKIN Rising	11		ns		24
t2b	A[17:2] Setup to HCLKIN Rising	6		ns		24
t2c	SIDLE#, A[17:2] Hold from HCLKIN Rising	2		ns		24
t2c1	CMDV# Hold from HCLKIN Rising	2.6		ns		24
t2d	CMDV#, SIDLE#, A[17:2] Valid Delay from HCLKIN Rising	3	10	ns		25
t2e	LGNT# Setup to HCLKIN Rising	11		ns		24
t2f	LGNT# Hold from HCLKIN Rising	2		ns		24
t2g	LREQ# Valid Delay from HCLKIN Rising	2	9	ns		25
t2h	SIDLE# Driven Valid After CPURST is Driven High	2		HCLKIN		22

6.3.3 SYSTEM POWER MANAGEMENT TIMINGS

AC Characteristics (V_{DD} = 5V ± 5%, T_{CASE} = 0 to 85°C)

Symbol	Parameter	Min	Max	Units	Notes	Fig
	SMI#					
t3a	Valid Delay from HCLKIN	2	9	ns		25
t3b	Active Pulse Width	3		HCLKIN		23
t3c	Inactive Pulse Width	4		HCLKIN		23
	EXTSMI#					
t3d	Active Pulse Width	2		HCLKIN		23
t3e	Inactive Pulse Width	4		HCLKIN		23
t3f	Valid Setup to HCLKIN	6		ns		24
t3g	Valid Hold from HCLKIN	2		ns		24
	STPCLK#					
t3h	Valid Delay from HCLKIN	2	10	ns		25
t3i	STPCLK# Inactive Pulse Width	5		HCLKIN		23

6.3.4 ISA BUS AND X-BUS TIMINGS

ISA Bus and X-Bus AC Characteristics (V_{DD} = 5V ± 5%, T_{CASE} = 0 to 85°C)

Symbol	Parameter	Min	Max	Units	Type	Size	Notes	Fig
	IB AS MASTER TIMINGS							
	BALE							
t4a	BALE Pulse Width	52		ns	M,I/O	8,16		26,27,28,29
t4b	BALE Driven Active from MEMx#, IOx# Inactive	44		ns	M,I/O	8,16		26,27,28,29
	LA[23:17]							
t5a	LA[23:17] Valid Setup to BALE Inactive	150		ns	M	8,16	7	26,27
t5b	LA[23:17] Valid Hold from BALE Inactive	26		ns	M	8,16		26,27
t5c	LA[23:17] Valid Setup to MEMx# Active	150		ns	M	16		27
t5d	LA[23:17] Valid Setup to MEMx# Active	173		ns	M	8		26
t5e	LA[23:17] Invalid from MEMx# Active	39		ns	M	16		27
t5f	LA[23:17] Invalid from MEMx# Active	39		ns	M	8		26
	SA[19:0], SBHE#							
t6a	SA[19:0], SBHE# Valid Setup to MEMx# Active	34		ns	M	16	13	27
t6b	SA[19:0], SBHE# Valid Setup to IOx# Active	100		ns	I/O	16		29
t6c	SA[19:0], SBHE# Setup to MEMx#, IOx# Active	100		ns	M,I/O	8		26,28
t6d	SA[19:0], SBHE# Valid Setup to BALE Inactive	37		ns	M,I/O	8,16	13	26,27,28,29
t6e	SA[19:0], SBHE# Valid Hold from MEMx#, IOx# Inactive	41		ns	M,I/O	8,16		26,27,28,29

ISA Bus and X-Bus AC Characteristics (V_{DD} = 5V ± 5%, T_{CASE} = 0 to 85°C) (Continued)

Symbol	Parameter	Min	Max	Units	Type	Size	Notes	Fig
	MEMR#, MEMW#, IOR# and IOW#							
t7a	MEMx# Active Pulse Width (std)	225		ns	M	16		27
t7b	IOx# Active Pulse Width (std)	160		ns	I/O	16		29
t7c	MEMx# Active Pulse Width (nws)	105		ns	M	16	1	27
t7d	MEMx# or IOx# Active Pulse Width (std)	520		ns	M,I/O	8		26,28
t7e	MEMx# or IOx# Active Pulse Width (nws)	160		ns	M,I/O	8	1	26,28
t7f	MEMx# Inactive Pulse Width	103		ns	M	16		27
t7g	MEMx# Inactive Pulse Width	163		ns	M	8		26
t7h	IOx# Inactive Pulse Width	163		ns	I/O	8,16		28,29
t7i	MEMx#, IOx# Driven Inactive from IOCHRDY Active	120		ns	M,I/O	8,16		26,27,28,29
	SMEMR# and SMEMW#							
t8a	SMEMR# & SMEMW# Propagation Delay from MEMR# and MEMW#		5	ns	M	8,16		26,27
	Read Data							
t9a	Read Data Driven from MEMR#, IOR# Active	0		ns	M,I/O	8,16		26,28,29
t9b	Read Data Valid Setup to MEMR#, IOR#	20		ns	M,I/O	8,16		26,27,28
t9c	Read Data Valid Hold from MEMR#, IOR# Inactive	0		ns	M,I/O	8,16		26,27,28,29
t9d	Read Data Tri-stated from MEMR# and IOR# Inactive		41	ns	M,I/O	8,16		26,27,28,29
	Write Data							
t10a	Write Data Valid Setup to MEMW#, IOW# Active	30		ns	M,I/O	8,16		26,27,28,29
t10b	Write Data Valid Hold from MEMW#, IOW# Inactive	45		ns	M,I/O	8,16		26,27,28,29
t10c	Write Data Tri-Stated from MEMW#, IOW# Inactive		75	ns	M,I/O	8,16		26,27,28,29
t10d	Write Data Driven Valid after Read MEMR#, IOR# Inactive	41		ns	M,I/O	8,16		26,27,28,29

ISA Bus and X-Bus AC Characteristics (V_{DD} = 5V ± 5%, T_{CASE} = 0 to 85°C) (Continued)

Symbol	Parameter	Min	Max	Units	Type	Size	Notes	Fig
	MEMCS16#							
t11a	MEMCS16# Driven Active from LA[23:17] Valid		94	ns	M	16		27
t11b	MEMCS16# Inactive from LA[23:17] Valid		91	ns	M	8		26,27
t11c	MEMCS16# Valid Hold from LA[23:17] Invalid	0		ns	M	16		27
t11d	MEMCS16# Driven Active from SA[19:2] Valid		35	ns	M	16		27
	IOCS16#							
t12a	IOCS16# Driven Active from Valid SA[19:0]		123	ns	I/O	16		29
t12b	IOCS16# Inactive from Valid SA[19:0]		91	ns	I/O	8		28,29
t12c	IOCS16# Valid Hold from SA[19:0] Invalid	0		ns	I/O	16		29
t12d	IOCS16# Driven Active from IOx Active		75	ns	I/O	16		29
	ZEROWS#							
t13a	ZEROWS# Driven Active from MEMx# Active		27	ns	M	16		27
t13b	ZEROWS# Driven Active from MEMx#, IOx# Active		80	ns	M,I/O	8		26,28
t13c	ZEROWS# Driven Active from LA[23:17] Valid		180	ns	M	16		27
t13d	ZEROWS# Driven Active from LA[23:17] Valid		300	ns	M	8		26
t13e	ZEROWS# Driven Active from SA[19:0], SBHE# Valid		80	ns	M	16		27
t13f	ZEROWS# Driven Active from SA[19:0], SBHE# Valid		200	ns	M,I/O	8		26,28
	AEN							
t14a	AEN Valid Setup to IOx# Driven Active	111		ns	I/O	8,16		28,29
t14b	AEN Valid Setup to BALE Driven Inactive	111		ns	I/O	8,16		28,29
t14c	AEN Valid Hold from IOx# Driven Inactive	41		ns	I/O	8,16		28,29
	IOCHRDY							
t15a	IOCHRDY Driven Valid from MEMx#, IOx# Active		78	ns	M,I/O	16		27,29
t15b	IOCHRDY Driven Valid from MEMx#, IOx# Active		366	ns	M,I/O	8		26,28
t15e	IOCHRDY Inactive Pulse Width	120	15.6	ns	M,I/O	8,16		26,28,29

ISA Bus and X-Bus AC Characteristics (V_{DD} = 5V ± 5%, T_{CASE} = 0 to 85°C) (Continued)

Symbol	Parameter	Min	Max	Units	Type	Size	Notes	Fig
	IB AS SLAVE TIMINGS							
	LA[23:17]							
t16a	LA[23:17] Valid Setup to MEMx# Active	23		ns	M	16		30
t16b	LA[23:17] Invalid from MEMx# Active	28		ns	M	16		30
	SA[19:0],SBHE#							
t17a	SA[19:0],SBHE# Setup to MEMx# Active	23		ns	M	16		30
t17b	SA[19:0],SBHE# Setup to IOx# Active	89		ns	I/O	8		31
t17c	SA[19:0],SBHE# Valid Hold from MEMx#, IOx# Inactive	30		ns	M,I/O	8,16		30,31
	MEMR#, MEMW#, IOR#, IOW#							
t18a	MEMx# Active Pulse Width	214		ns	M	16		30
t18b	IOx# Active Pulse Width	509		ns	I/O	8		31
t18c	MEMx# Inactive Pulse Width	92		ns	M	16		30
t18d	IOx# Inactive Pulse Width	152		ns	I/O	8		31
	Read Data							
t19a	Read Data Valid from IOCHRDY Active		69	ns	M,I/O	8,16		30,31
t19b	Read Data Valid from IOR# Active		69	ns	I/O	8	11	31
t19c	Read Data Valid Hold from MEMR#, IOR# Inactive	0		ns	M,I/O	8,16		30,31
t19d	Read Data Tri-State from MEMR#, IOR# Inactive		30	ns	M,I/O	8,16		30,31
	Write Data							
t20a	Write Data Valid Setup to MEMW#, IOW# Active	−54		ns	M,I/O	8,16		30,31
t20b	Write Data Valid Hold from MEMW#, IOW# Inactive	14		ns	M,I/O	8,16		30,31
	MEMCS16#							
t21a	MEMCS16# Driven Active from Valid LA[23:17]		65	ns	M	16		30
t21b	MEMCS16# Float from Valid LA[23:17]		31	ns	M	16		30
t21c	MEMCS16# Valid Hold from LA[23:17] Invalid	0		ns	M	16		30
	IOCHRDY							
t22a	IOCHRDY Inactive from MEMx#, IOx# Active		25	ns	M,I/O	8,16		30,31
t22b	IOCHRDY Float from IOCHRDY Rising		70	ns	M,I/O	8,16	4	30,31
t22c	IOCHRDY Inactive Pulse Width	120	2.5	μs	M,I/O	8,16		30,31

ISA Bus and X-Bus AC Characteristics (V_{DD} = 5V ± 5%, T_{CASE} = 0 to 85°C) (Continued)

Symbol	Parameter	Min	Max	Units	Type	Size	Notes	Fig
	INTERRUPT AND NMI TIMINGS							
	NMI Timing							
t23a	SERR#, IOCHK# Active to NMI Driven Active		200	ns				32
	Interrupt Timing							
t24a	IRQ Inactive Pulse Width	100		ns				33
	ISA BUS MASTER TIMINGS							
	DACK#							
t26a	DACK# Inactive from DREQ Inactive	240		ns				34
	Tri-Stating and Driving the Bus							
t27a	IB Tri-States Address, Data, and Control Signals from DACK# Active		30	ns				34
t27b	IB Drives Address, Data, and Control Signals from DACK# Inactive	71		ns				34
	SMEMR# and SMEMW#							
t28a	SMEMR# & SMEMW# Valid from MEMR# and MEMW# Valid		20	ns				34
	DATA SWAP LOGIC TIMING (ISA Master to ISA Slave)							
t29a	SD[7:0] to SD[15:8] Propagation Delay		15	ns				35
t29b	SD[15:8] to SD[7:0] Propagation Delay		15	ns				35
t29c	IB Drives Data Bus from IOR#, IOW#, MEMR# or MEMW# Active		20	ns			2	35
t29d	IB Tri-States Bus from IOR#, MEMR#, or SMEMR# Inactive	5	20	ns			2,3	35
t29e	IB Tri-States Bus from IOW#, MEMW#, or SMEMW# Inactive	15	60	ns			2,3	35
	DMA COMPATIBLE TIMINGS							
	DREQ							
t30a	DREQ Active Hold from IOR# Active		558	ns			5	37
t30b	DREQ Active Hold from IOW# Active		315	ns			5	36
	DACK#							
t31a	DACK# Active to IOR# Active	73		ns				37
t31b	DACK# Active to IOW# Active	312		ns				36
t31c	DACK# Active Hold from IOR# Inactive	105		ns				37
t31d	DACK# Active Hold from IOW# Inactive	161		ns				36

ISA Bus and X-Bus AC Characteristics (V_{DD} = 5V ± 5%, T_{CASE} = 0 to 85°C) (Continued)

Symbol	Parameter	Min	Max	Units	Type	Size	Notes	Fig
	AEN and BALE							
t32a	AEN Active to IOx# Active	111		ns				36,37
t32b	AEN and BALE Inactive from IOx# Inactive	41		ns				36,37
	LA[23:19], SA[19:0], SBHE#							
t33a	LA[23:19],SA[19:0], SBHE# Valid Setup to MEMx# Active	99		ns				36,37
t33b	LA[23:19],SA[19:0], SBHE# Valid Hold from MEMx# Inactive	51		ns				36,37
	MEMR#, MEMW#, IOR#, IOW#							
t34a	IOW# and MEMW# Active Pulse Width	474		ns				36,37
t34b	MEMR# Active Pulse Width	520		ns				36
t34c	IOR# Active Pulse Width	769		ns				37
t34d	IOW# Inactive Pulse Width (continuous)	469		ns				36
t34e	IOR# Inactive Pulse Width (continuous)	167		ns				37
t34f	IOR# Active to MEMW# Active	235		ns				37
t34g	MEMR# Active to IOW# Active	0		ns				36
t34h	MEMR# Active Hold from IOW# Inactive	50		ns				36
t34i	IOR# Active Hold from MEMW# Inactive	50		ns				37
t34j	MEMx# Active Hold from IOCHRDY Active	120		ns				36,37
	SMEMR# and SMEMW#							
t35a	SMEMR# and SMEMW# Valid from MEMR# and MEMW# Valid		5	ns				36,37
	Read Data							
t36a	Read Data Valid from IOR# Active		237	ns				37
t36b	Read Data Valid Hold from IOR# Inactive	0		ns				37
t36c	Read Data Float from IOR# Inactive		61	ns				37
	Write Data							
t37a	Write Data Valid Setup to IOW# Inactive	252		ns				36
t37b	Write Data Valid Hold from IOW# Inactive	36		ns				36

ISA Bus and X-Bus AC Characteristics (V_{DD} = 5V ± 5%, T_{CASE} = 0 to 85°C) (Continued)

Symbol	Parameter	Min	Max	Units	Type	Size	Notes	Fig
	DATA SWAP LOGIC TIMING (ISA to ISA Transaction)							
t38a	SD[7:0] to SD[15:8] Propagation Delay		15	ns				38
t38b	SD[15:8] to SD[7:0] Propagation Delay		15	ns				38
t38c	IB Drives Data Bus from IOR# or MEMR# Active		20	ns			2	38
t38d	IB Tri-States Bus from IOR# or MEMR# Inactive		20	ns			2	38
	TC							
t39a	TC Active Setup to IOx# Inactive	511		ns			6	36,37
t39b	TC Active Hold from IOx# Inactive	71		ns			6	36,37
t39h	TC Pulse Width	700		ns				36,37
	IOCHRDY							
t40b	IOCHRDY Valid from MEMx# Active		315	ns				36,37
t40c	IOCHRDY Inactive Pulse Width	125		ns				36,37
	ISA REFRESH TIMINGS							
	REFRESH#							
t62a	REFRESH# Active Setup to MEMR# Active	120		ns				39,40
t62b	REFRESH# Active Hold from MEMR# Inactive	31	218	ns				8,40
t62c	REFRESH# Driven Active to SA[15:0] Valid	11		ns				8,40
t62d	REFRESH# Active Hold from SA[15:0] Invalid	11		ns				8,40
	AEN							
t63a	AEN Driven Active to MEMR# Active	11		ns				39,40
t63b	AEN Hold from MEMR# Inactive	11		ns				39,40
	SA[15:0]							
t64a	SA[15:0] Valid Setup to MEMR# Active	81		ns				39,40
t64b	SA[15:0] Valid Hold from MEMR# Inactive	36		ns				39,40
t64c	SA[15:0] Valid Float from MEMR# Inactive	45	120	ns			8	40

ISA Bus and X-Bus AC Characteristics (V_{DD} = 5V ± 5%, T_{CASE} = 0 to 85°C) (Continued)

Symbol	Parameter	Min	Max	Units	Type	Size	Notes	Fig
	MEMR#, SMEMR#							
t65a	MEMR# Active Pulse Width	225		ns				39,40
t65b	MEMR# Tri-state from MEMR# Inactive	45	120	ns				39,40
t65c	MEMR# Driven Inactive from IOCHRDY Active	120		ns				39,40
t65d	SMEMR# Propagation Delay from MEMR#		5	ns				39
	IOCHRDY							
t66a	IOCHRDY Inactive from MEMR# Active		76	ns				39,40
t66b	IOCHRDY Valid from MEMR# Active		76	ns				39,40
t66c	IOCHRDY Active to Inactive	120		ns				39,40
	IB Driving Bus from REFRESH#							
t67a	IB Drives Control and Address from REFRESH# Active	5		ns			8	40
	IB AND ISA MASTER ACCESSES TO THE X-BUS							
	BIOSCS#, KBCCS#, and RTCCS#							
t68a	CS# Driven Active from SA[19:0], LA[23:17] Valid		25	ns				41
t68b	CS# Driven Inactive from SA[16:0], LA[23:17] Invalid		25	ns				41
	XBUSTR# and XBUSOE#							
t69a	XBUSTR# Active from IOR#, MEMR# Active		17	ns				41
t69b	XBUSOE# Active from IOx#, MEMx# Active		29	ns				41
t69c	XBUSTR# Active Setup to XBUSOE# Active	3	12	ns				41
t69d	XBUSOE# Inactive from IOx#, MEMx# Inactive	35	60	ns			9	41
t69e	XBUSTR# Inactive from IOR#, MEMR# Inactive	45	100	ns			9	41
t69f	XBUSOE# Setup to XBUSTR# Inactive	10	45	ns			9	41
t69g	XBUSOE# Inactive from SA[16:0] and LA[23:17]		25	ns			10	41
t69h	XBUSTR# Inactive from IOR#, MEMR# Inactive	15	60	ns			10	41

ISA Bus and X-Bus AC Characteristics (V_{DD} = 5V ± 5%, T_{CASE} = 0 to 85°C) (Continued)

Symbol	Parameter	Min	Max	Units	Type	Size	Notes	Fig
	DMA ACCESSES TO X-BUS							
	XBUSTR#							
t70a	XBUSTR# Active from DACKx# Active		25	ns			12,14	42
t70b	XBUSTR# Inactive from DACKx# Inactive	10	65	ns			12	42
	MISCELLANEOUS X-BUS TIMINGS							
	Mouse Timing Support							
t71a	IRQ12/M and IRQ1 Minimum Active Pulse Width (for Mouse Function and Keyboard)	180		ns				43
	Coprocessor Error Support							
t73a	IGNNE# Active from IOW# Active from Port F0h Access		220	ns				43
t73b	IGNNE# Inactive from FERR# Inactive		150	ns				43
	Real Time Clock Timing (RTCALE)							
t75a	RTCALE Pulse Width	200	300	ns				44
t75b	RTCALE Active from IOW# Active		70	ns				44
	Speaker Timing							
t76a	SPKR Valid Delay from OSC Rising		200	ns				45

NOTES:
1. No-wait-state (ZEROWS#) asserted.
2. This applies to the byte lane that the data has been swapped to.
3. Data is tri-stated from the standard memory commands (SMEMR# or SMEMW#), when they are generated.
4. This specification includes both the time the IB drives IOCHRDY active and the time it takes the IB to float IOCHRDY.
5. This applies to the last cycle of a demand mode DMA transfer.
6. Output from IB.
7. 36 ns has been added to the ISA spec to meet ZEROWS# setup requirements.
8. This applies to ISA Master initiated refresh only.
9. IB as a master cycles only.
10. ISA master cycles only.
11. This applies to the IB cycles that IOCHRDY is not driven low.
12. This applies to all DACK# signals.
13. 56 ns has been added to the ISA spec to meet MEMCS16# setup requirements. ISA devices are not suppose to use the SA address as part of their MEMCS16# decode. However, some devices do use SA as part of MEMCS16# decode.
14. X-Bus read
15. For back-to-back ''sub cycles'' generated as a result of byte assembly or disassembly, this spec is 34 ns.

6.3.5 AC TEST LOADS

Table 23. AC Test Loads

Capacitive Load	Signals
200 pF	REFRESH#, TC, SD[15:0], SA[19:0], SBHE#, LA[23:17], I0CS16#, MASTER#, MEMCS16#, MEMR#, MEMW#, SMEMR#, SMEMW#, IOR#, IOW#, AEN, BALE, IOCHRDY, ZEROWS#, RSTDRV, SYSCLK
120 pF	DACK#[7:5,3:0]
50 pF	SPKR, INTR, NMI, BIOSCS#, KBCCS#, RTCCS#, RTCALE, XBUSTR#, XBUSOE#, IGNNE#

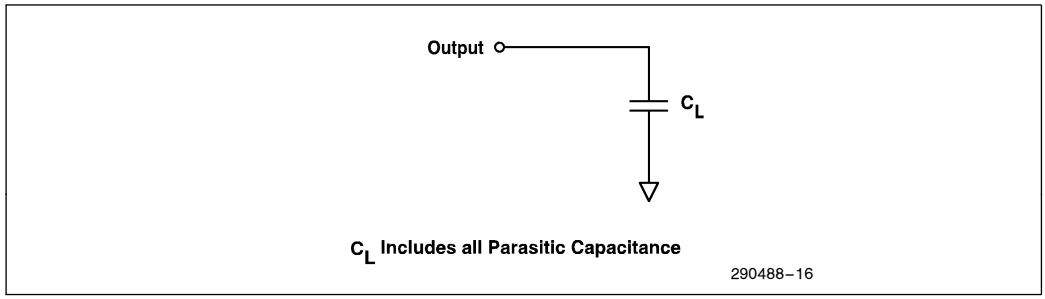


Figure 20. Test Load

6.3.6 AC TIMING WAVEFORMS

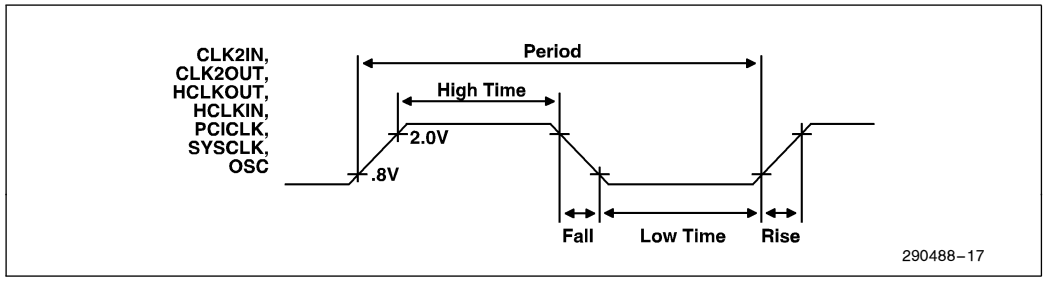


Figure 21. Clock Timing

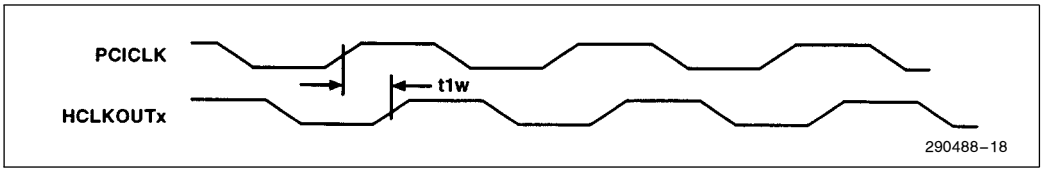


Figure 22. PCICLK-to-HCLKOUT Skew Timing

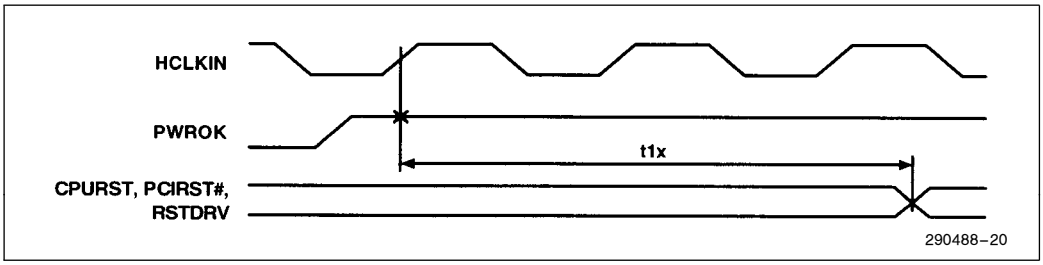


Figure 23. Reset Inactive after PWROK

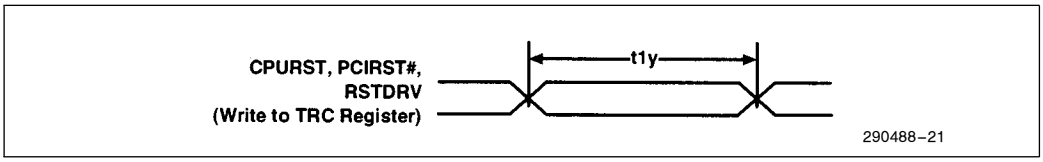


Figure 24. Reset Active Pulse Width

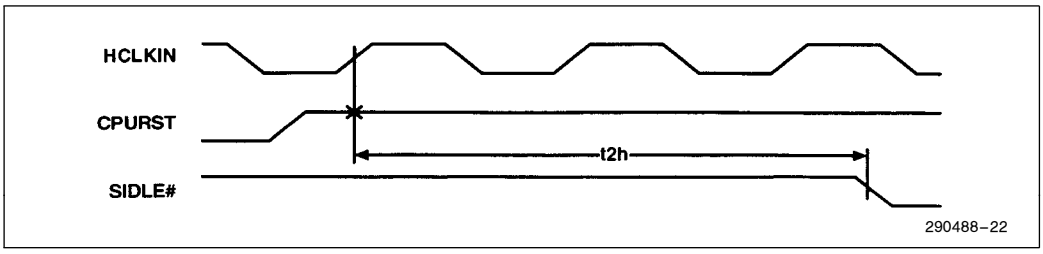


Figure 25. SIDLE# Active after CPURST

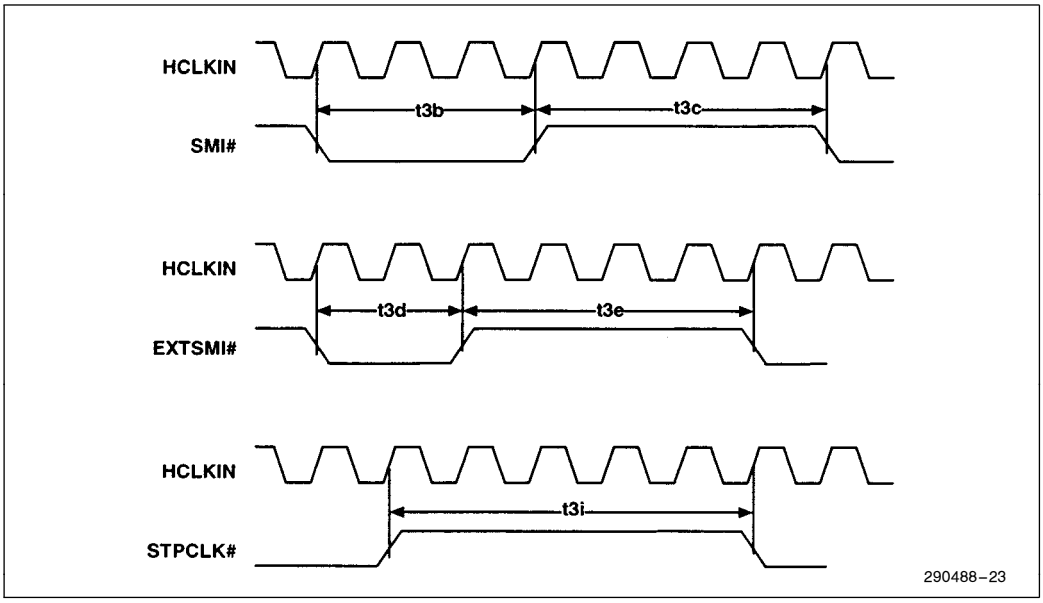


Figure 26. SMI#, EXTSMI#, and STPCLK# Timing

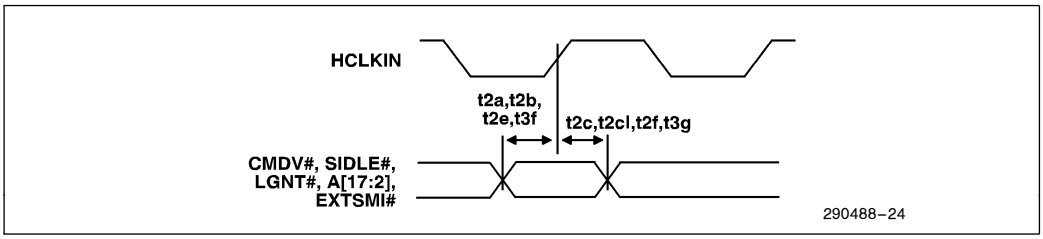


Figure 27. Input to HCLKIN Setup/Hold Times

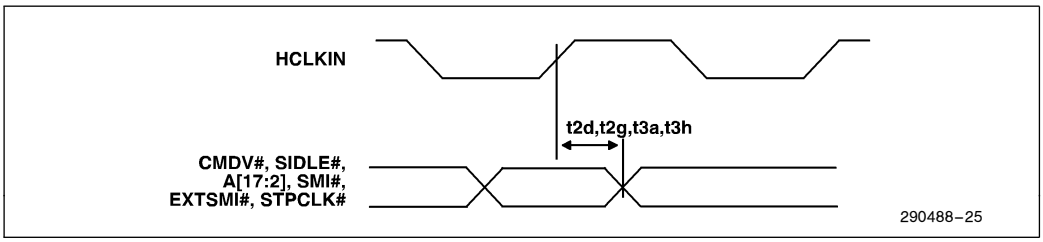


Figure 28. HCLKIN to Output Valid Delay

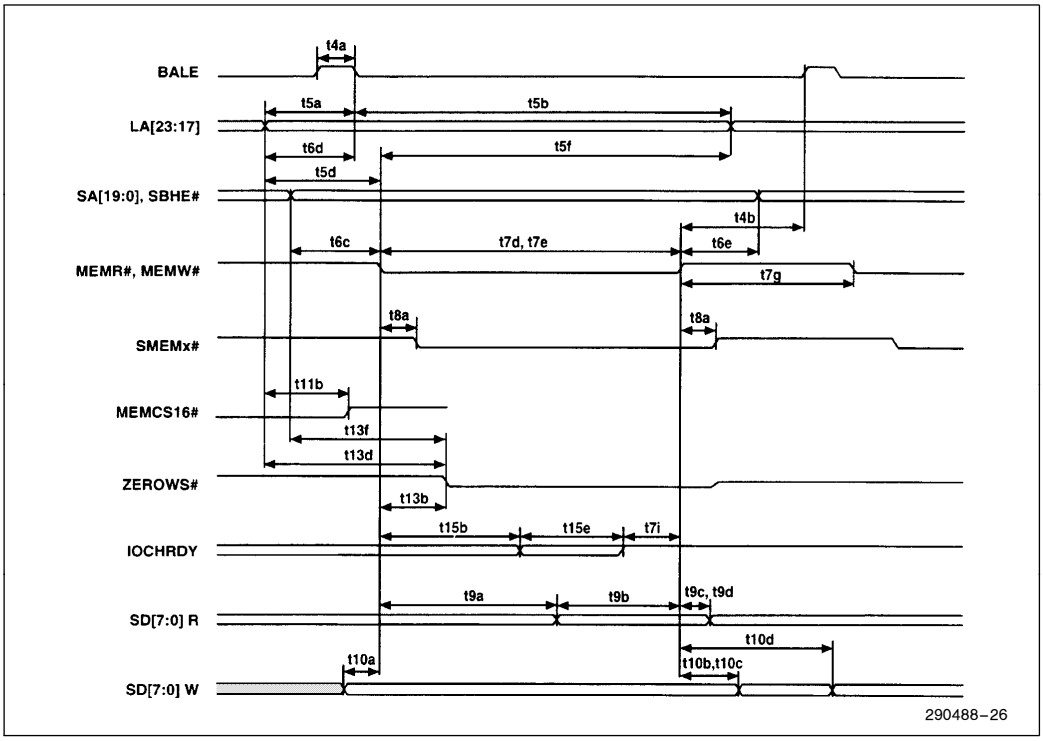


Figure 29. 8-Bit ISA Memory Slave Timing (IB as Master)

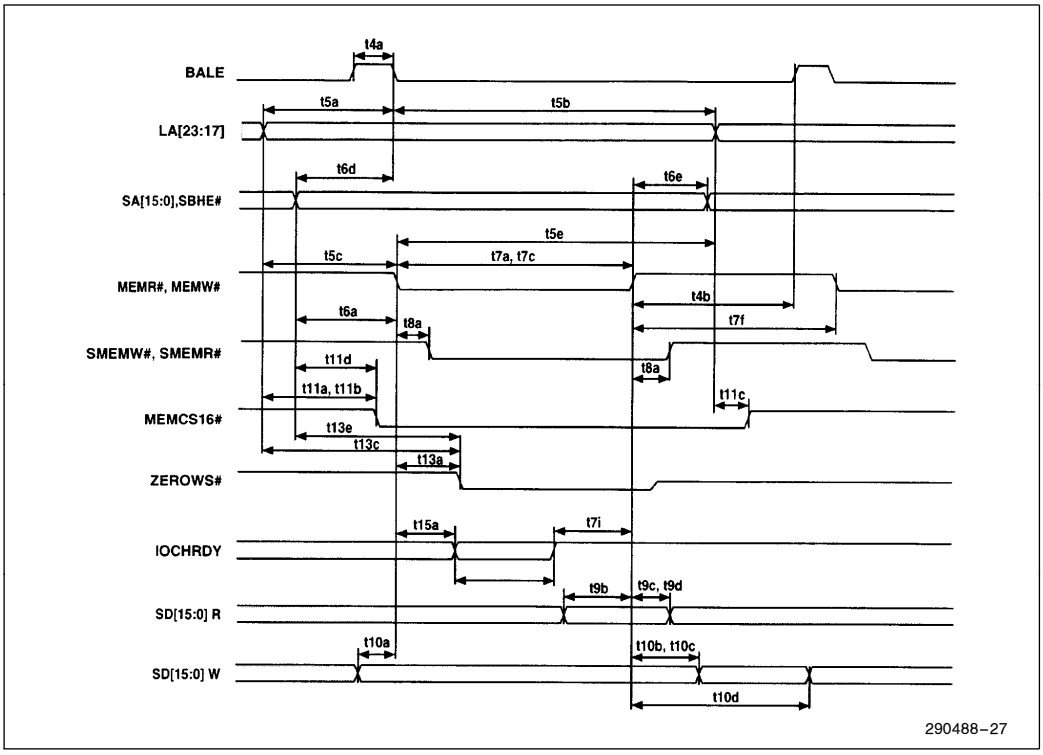


Figure 30. 16-Bit ISA Memory Slave Timing (IB as Master)

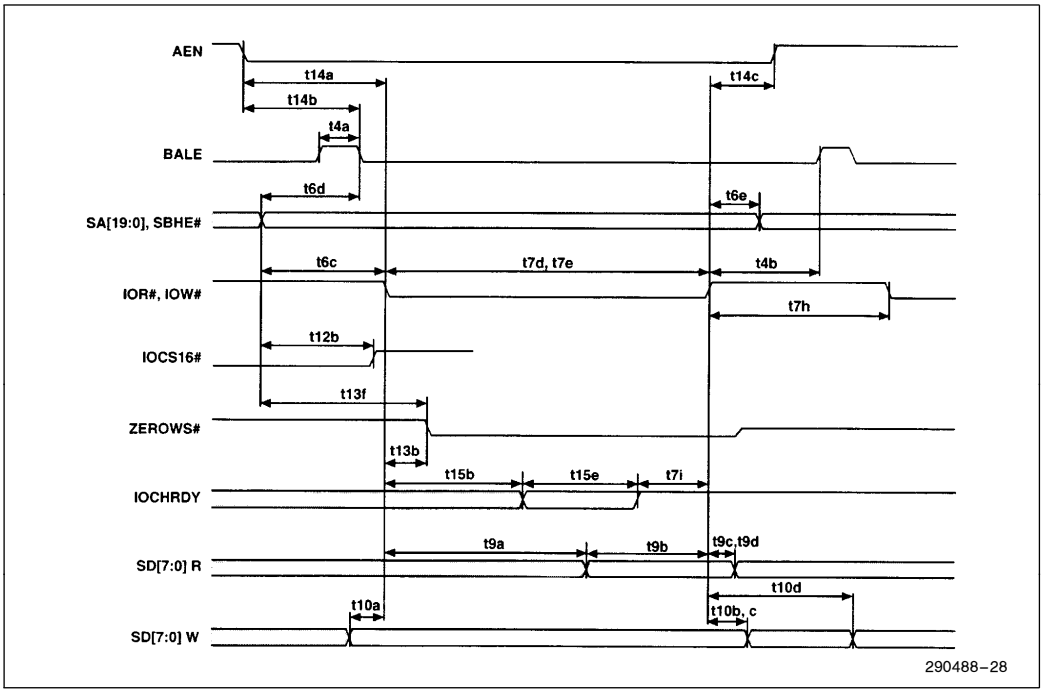


Figure 31. 8-Bit ISA I/O Slave Timing (IB as Master)

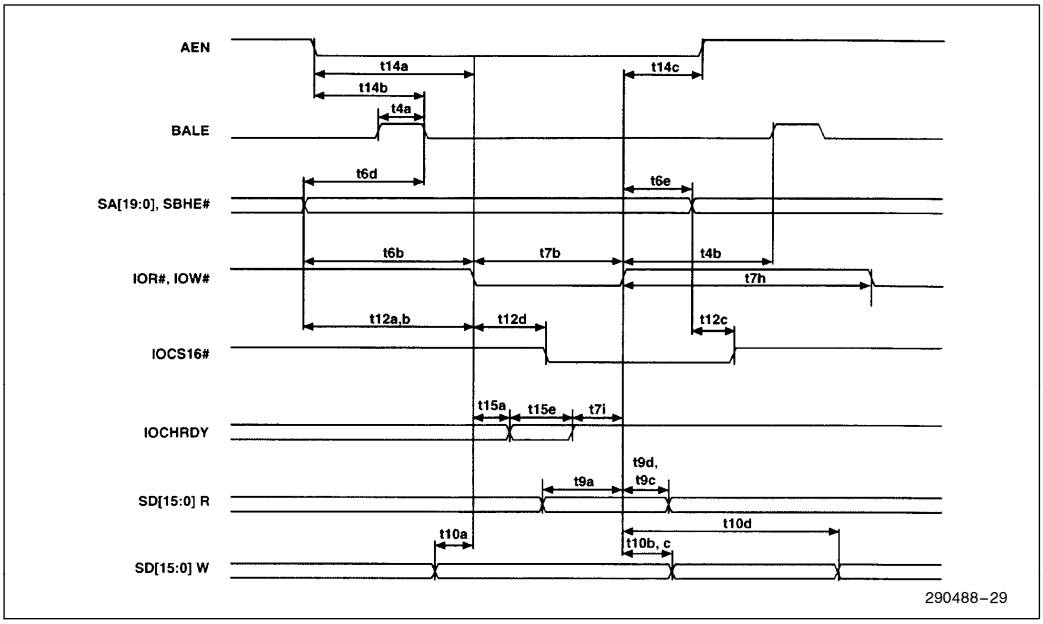


Figure 32. 16-Bit I/O Slave Timing (IB as Master)

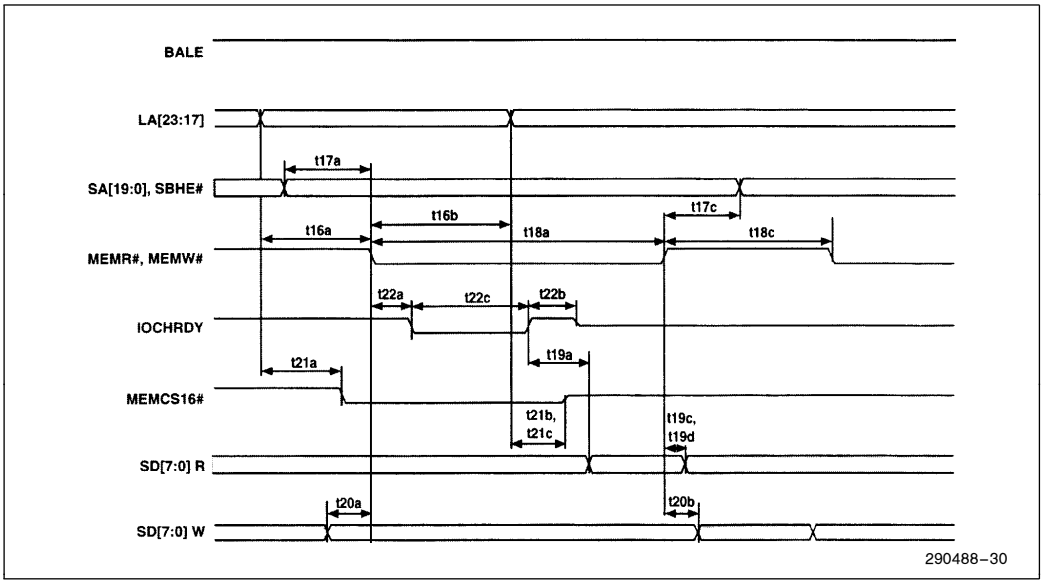


Figure 33. ISA Master Accessing PCI Memory Timing

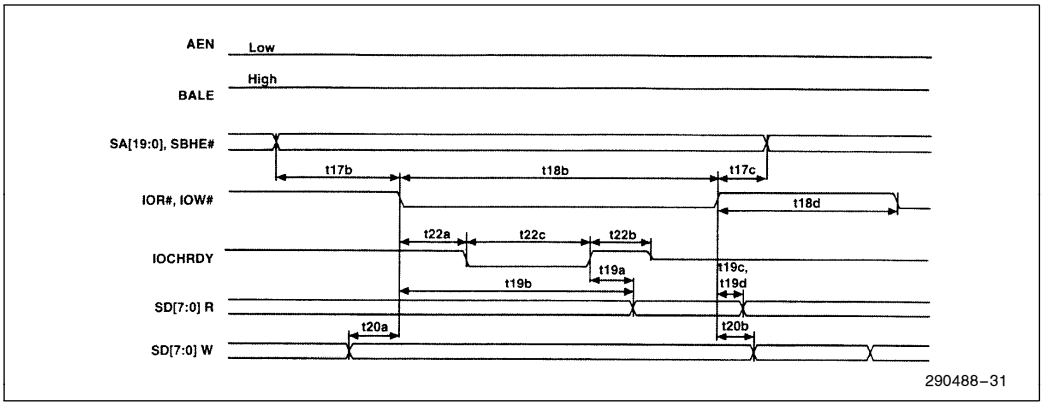


Figure 34. ISA Master Accessing IB Register Timing

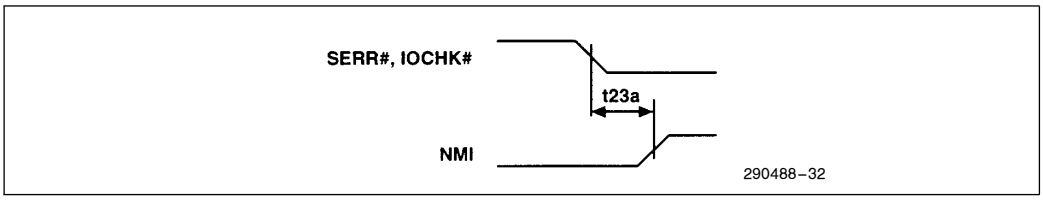


Figure 35. NMI Timing

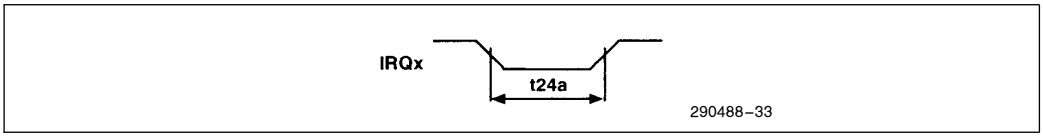


Figure 36. Interrupt Timing

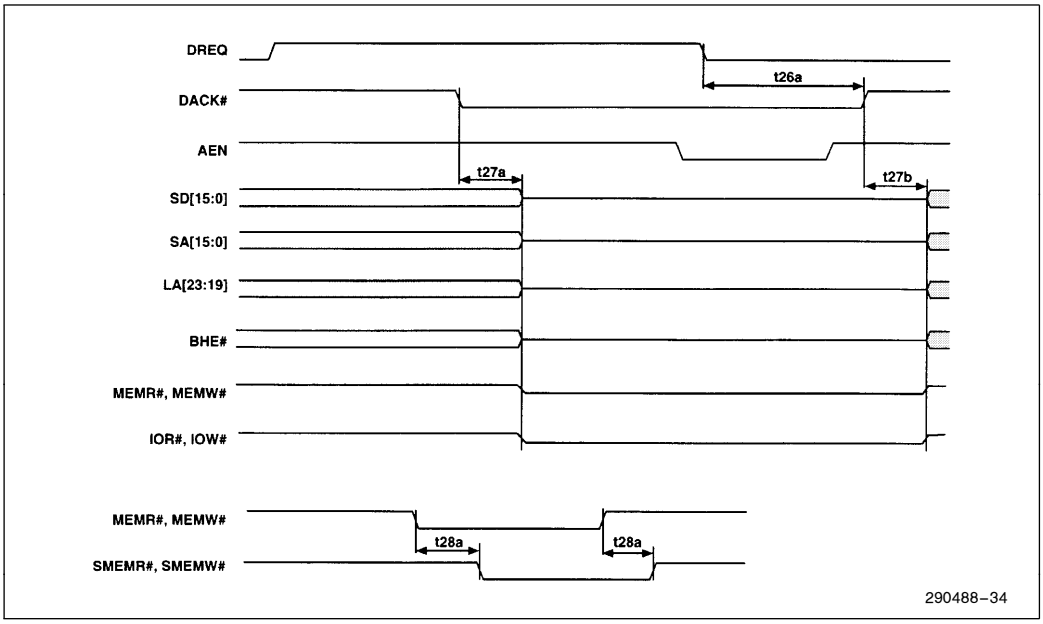


Figure 37. ISA Master Miscellaneous Timing

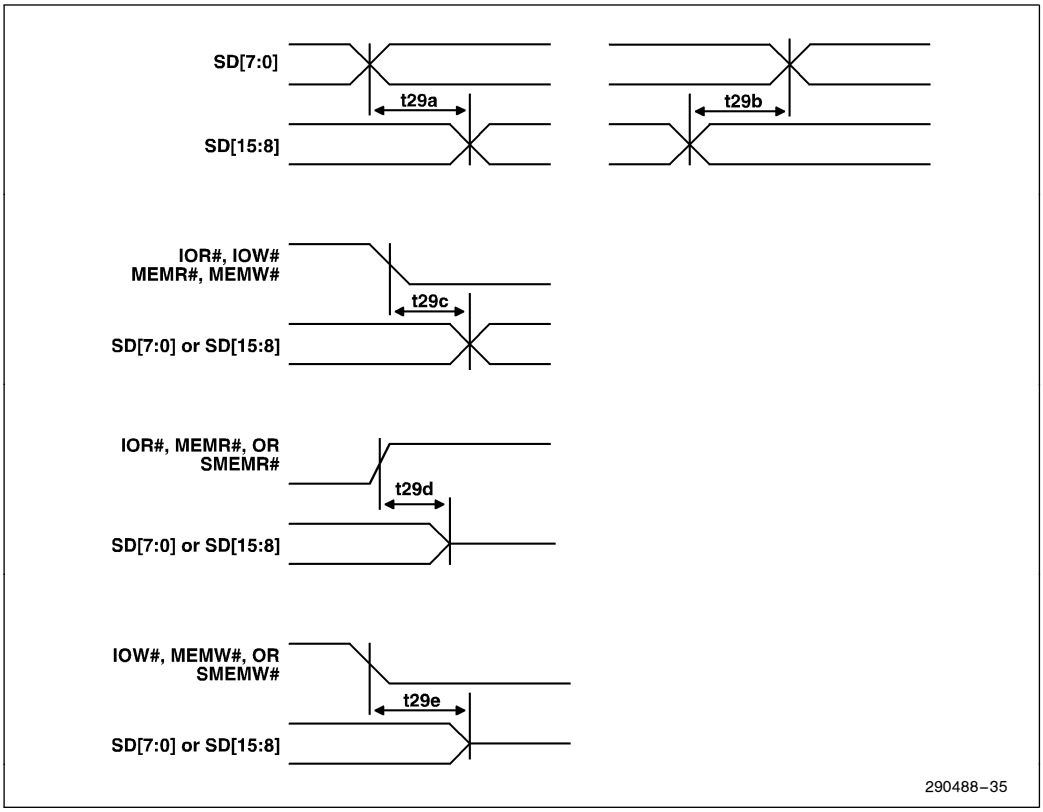


Figure 38. ISA Master Data Swap Timing

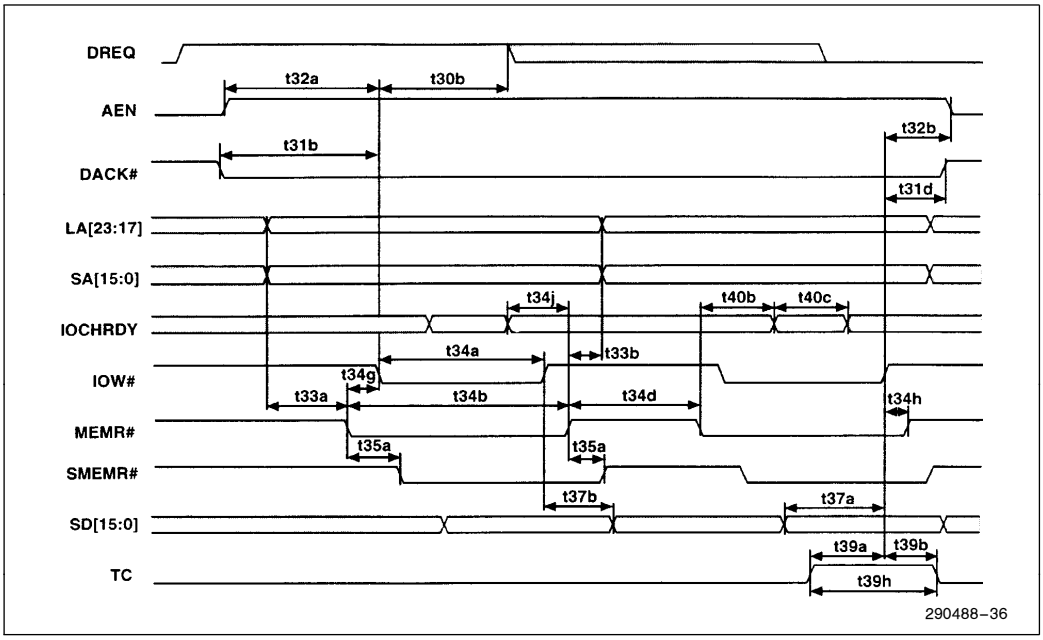


Figure 39. DMA Compatible Timing (Memory Read)

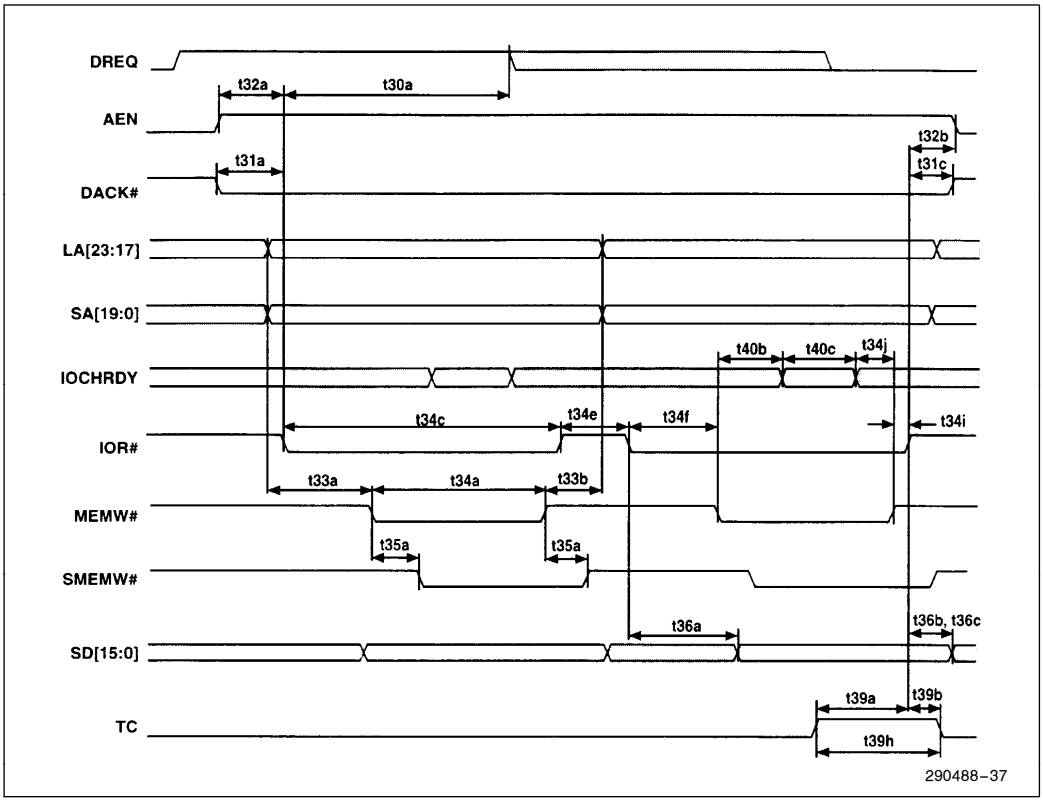


Figure 40. DMA Compatible Timing (Memory Write)

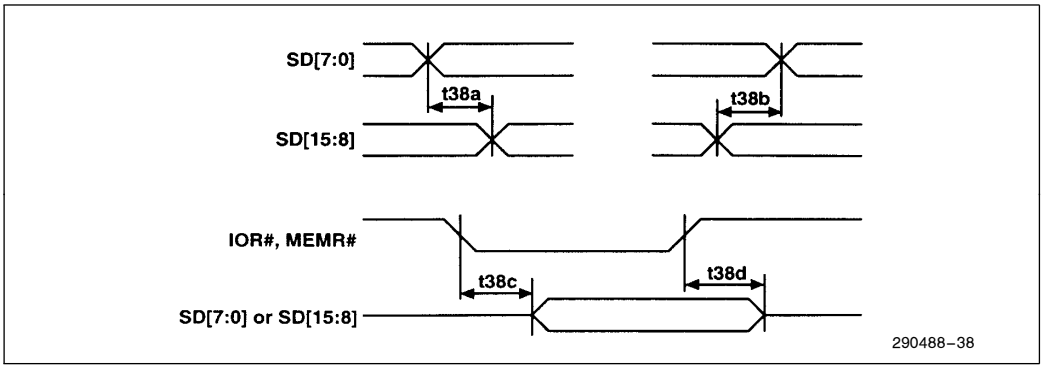


Figure 41. DMA Compatible Timing (Data Swap)

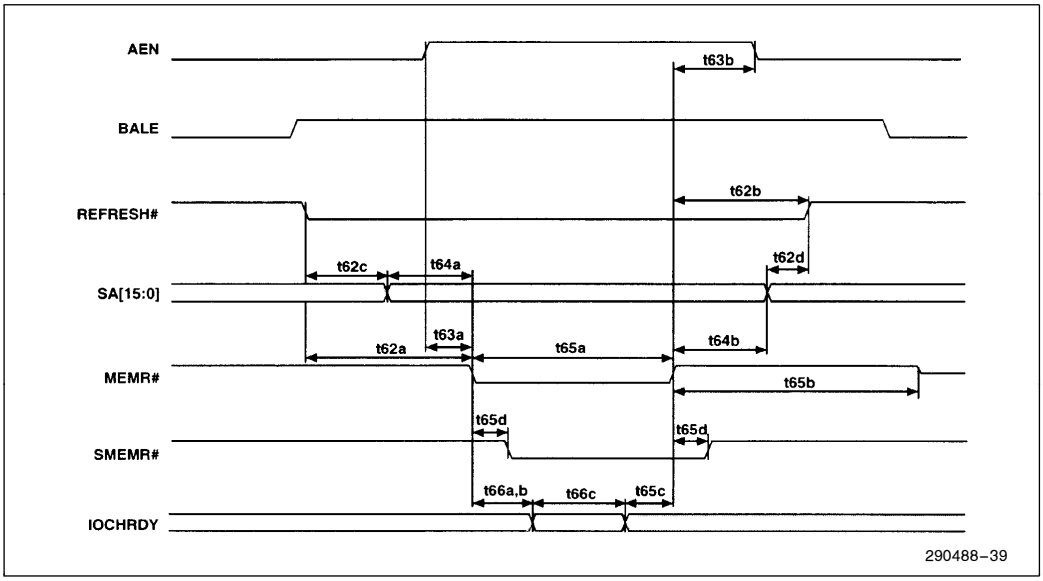


Figure 42. IB-Initiated Refresh Timing

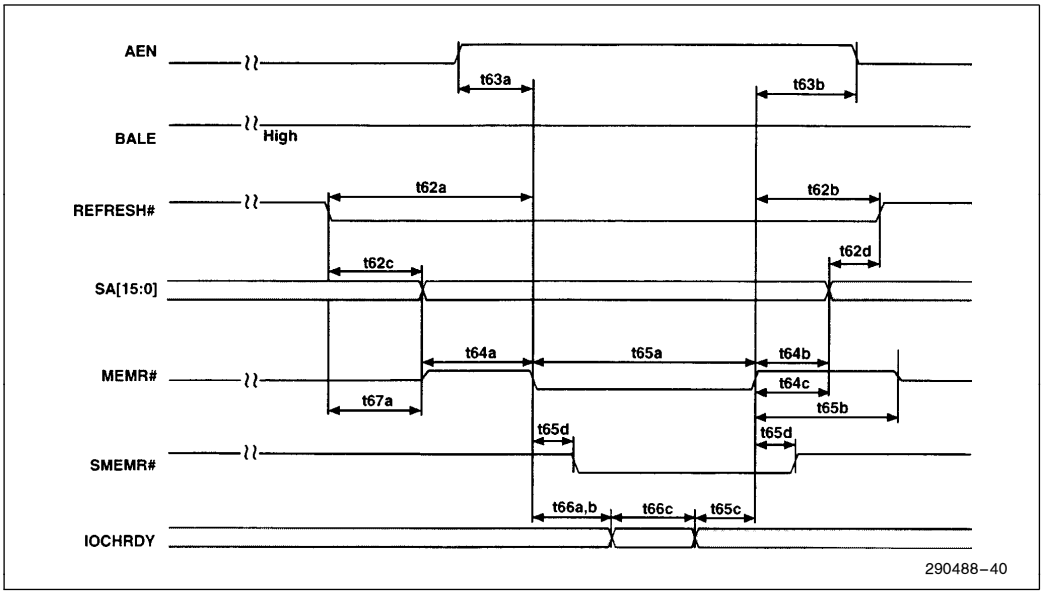


Figure 43. ISA Master-Initiated Refresh Timing

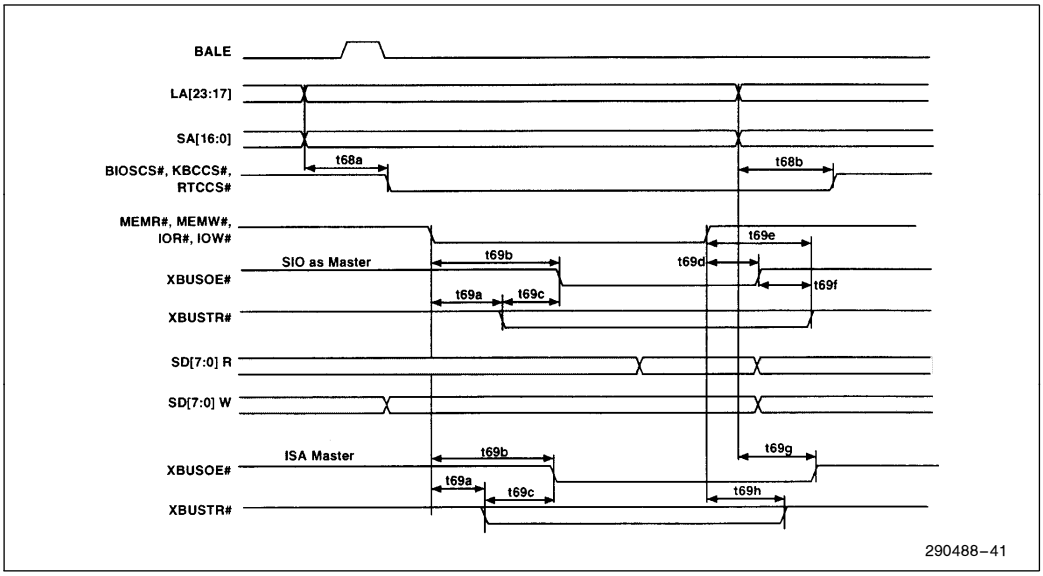


Figure 44. IB and ISA Master Access to X-Bus Timing

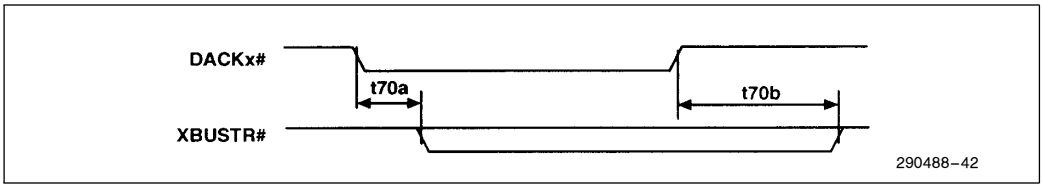


Figure 45. DMA Access to X-Bus Timing

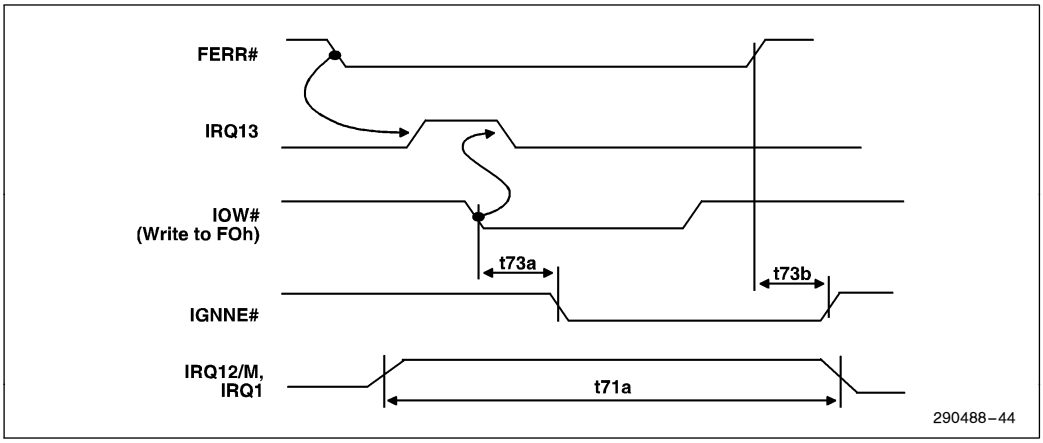


Figure 46. Coprocessor Error and Mouse Support Timing

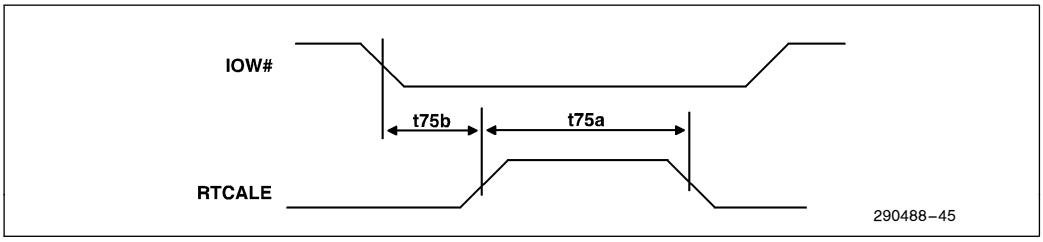


Figure 47. Real Time Clock Timing (RTCALE Generation)

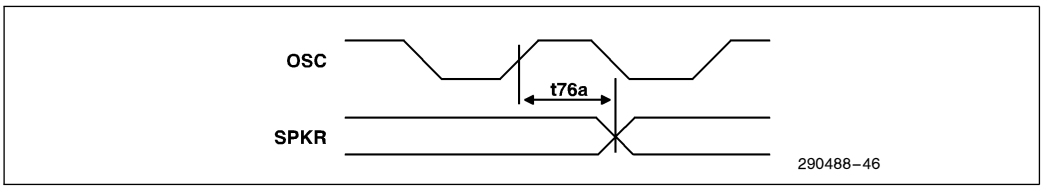


Figure 48. Speaker Timing

6.4 PSC AC Characteristics

This section provides the AC parameters and timing diagrams for the 82425EX PSC.

6.4.1 HOST CLOCK TIMING

AC Characteristics (V_{DD} = 5V ± 5%, T_{CASE} = 0 to 85°C)

Symbol	Parameter	Min	Max	Units	Notes	Fig
t1a	HCLKIN Period Stability		0.1%			
t1b	HCLKIN Period	30.0	40.0	ns		49
t1c	HCLKIN High Time	10.0		ns		49
t1d	HCLKIN Low Time	10.0		ns		49
t1e	HCLKIN Rise Time		2.0	ns		49
t1f	HCLKIN Fall Time		2.0	ns		49
t2a	CLK2IN Period Stability		0.1%			
t2b	CLK2IN Period	15	20.0	ns	Note 1	47

6.4.2 CPU INTERFACE TIMINGS

AC Characteristics (V_{DD} = 5V ±5%, T_{CASE} = 0 to 85 °C)

Symbol	Parameter	Min	Max	Units	Notes	Fig
t10a	HITM#, PCD, HLDA, BE[3:0], SMIACT#, SMI# Setup Time to HCLKIN Rising	10.0		ns		51
t10a1	ADS#, BLAST# Setup Time to HCLKIN Rising	12.0		ns		51
t10b	W/R#, M/IO#, D/C# Setup Time to HCLKIN Rising	14.0		ns		51
t10c	ADS#, HITM#, W/R#, M/IO#, D/C#, PCD, HLDA, BLAST#, BE[3:0]#, SMIACT#, SMI#, HLDA Hold Time from HCLKIN Rising	2.0		ns		51
t11a	HD[31:0], HDP[3:0] Setup Time to HCLKIN Rising	10.0		ns		51
t11b	HD[31:0], HDP[3:0] Hold Time from HCLKIN Rising	2.0		ns		51
t11c	HD[31:0], HDP[3:0] Output Enable from HCLKIN Rising	0.0	12.0	ns		54

AC Characteristics (V_{DD} = 5V ±5%, T_{CASE} = 0 to 85 °C) (Continued)

Symbol	Parameter	Min	Max	Units	Notes	Fig
t11d	HD[31:0], HDP[3:0] Valid Delay from HCLK Rising	3.0	14.0	ns		51
t11e	HD[31:0], HDP[3:0] Float Delay from HCLKIN Rising	0.0	12.0	ns		53
t12a	A[31,26:2] Setup Time to HCLKIN Rising	12.0		ns		51
t12b	A[31,26:2] Hold Time from HCLKIN Rising	2.0		ns		51
t12c	A[31,26:2] Output Enable from HCLKIN Rising	0.0	14.0	ns		54
t12d	A[31,26:2], BE[3:0]#, W/R# Valid Delay from HCLKIN Rising	3.0	14.0	ns		51
t12e	A[31,26:2], BE[3:0]#, W/R# Float Delay from HCLKIN Rising	0.0	12.0	ns		53
t13a	RDY#, BRDY# Rising Edge Valid Delay from HCLKIN Rising	3.0	16.0	ns		51
t13b	RDY#, BRDY# Falling Edge Valid Delay from HCLKIN Rising	3.0	16.0	ns		51
t14	AHOLD Valid Delay from HCLKIN Rising	3.0	11.0	ns		51
t15	EADS#, CPURST, HOLD Valid Delay from HCLKIN Rising	3.0	13.0	ns		52
t15a	KEN# Valid Delay from HCLKIN Rising	3.0	22.0	ns		51
t16a	INIT/SRESET High Pulse Width	16		HCLK		56
t16b	INIT/SRESET Valid Delay from HCLKIN Rising	3.0	13.0	ns		51
t17a	CPURST Setup Time to HCLKIN Rising	10.0		ns	Note 1	52
t17b	CPURST Hold Time to HCLKIN Rising	2.0		ns	Note 1	52
t17c	CPURST Pulse Width	3		HCLK	Note 2	56
t18a	LDEV# Setup Time to HCLKIN Rising	7.0				52
t18b	LDEV# Hold Time to HCLKIN Rising	2.0				52
t18c	LRDY# Early Setup Time to HCLKIN Rising	15.0				52
t18d	LRDY# Late Setup Time to HCLKIN Rising	7.0				52
t18e	LRDY# Hold Time to HCLKIN Rising	2.0				52
t18f	LRDY# to RDY# Propagation Delay	0.0	11.0			50

NOTES:
1. Synchronous Reset
2. Asynchronous Reset

6.4.3 SECOND LEVEL CACHE TIMING

AC Characteristics (V_{DD} = 5V ±5%, T_{CASE} = 0 to 85 °C)

Symbol	Parameter	Min	Max	Units	Notes	Fig
t20a	CI3E/CI3O2 Propagation Delay from A3	0.0	9.0	ns		57
t20b	CI3E/CI3O2 Valid Delay from HCLKIN Rising	3.0	13.0	ns		51
t21a	COE[1:0]# Falling Edge Valid Delay from HCLKIN Rising	3.0	13.0	ns		51
t21b	COE[1:0]# Rising Edge Valid Delay from HCLKIN Rising	3.0	13.0	ns		51
t22a	CWE[1:0]# Falling Edge Valid Delay from HCLKIN Rising	3.0	12.0	ns		51
t22b	CWE[1:0]# Rising Edge Valid Delay from HCLKIN Rising	3.0	13.0	ns		51
t22c	CWE[1:0]# Low Pulse Width	14.0		ns		56
t23	TWE# Valid Delay from HCLKIN Rising	3.0	14.0	ns		51
t24a	TAG[8:0] Valid Delay from HCLK	3.0	17.0	ns		
t24b	TAG[8:0] Setup Time to HCLKIN Rising	4.0		ns		52
t24c	TAG[8:0] Hold Time to HCLKIN Rising	7.0		ns		52

6.4.4 DRAM INTERFACE TIMING

AC Characteristics (V_{DD} = 5V ±5%, T_{CASE} = 0 to 85 °C)

Symbol	Parameter	Min	Max	Units	Notes	Fig
t30a	RAS[4:0]# Valid Delay from HCLKIN Rising	3.0	13.0	ns		51
t30b	RAS[4:0]# Pulse Width High	4		HCLK −5 ns		56
t31a	CAS[7:0]# Valid Delay from HCLKIN Rising	3.0	11.0	ns		51
t31b	CAS[7:0]# Pulse Width High	1		HCLK −2 ns		56
t32	WE# Valid Delay from HCLKIN Rising	3.0	17.0	ns		51
t33a	MA[10:0] Propagation Delay from A[26:5]	0.0	10.0	ns		57
t33b	MA[10:0] Row to Column Switching Delay	0.0	19.0	ns		57
t33c	MA[10:0] Valid Delay from HCLKIN Rising	3.0	15.0	ns		51

6.4.5 PCI TIMING

AC Characteristics (V_{DD} = 5V ±5%, T_{CASE} = 0 to 85 °C)

Symbol	Parameter	Min	Max	Units	Notes	Fig
	PCI Clock Timing					
t41a	Period	30.0		ns		49
t41b	PCLKIN High Time	8.0		ns		49
t41c	PCLKIN Low Time	8.0		ns		49
t41d	PCLKIN Rise Time		3.0	ns		49
t41e	PCLKIN Fall Time		3.0	ns		496
	PCI Interface Timing					
t50a	C/BE[3:0]#, FRAME#, TRDY#, IRDY#, STOP#, LOCK#, PAR, SERR#, DEVSEL# Valid Delay from PCLKIN Rising	2.0	11.0	ns		51
t50b	C/BE[3:0]#, FRAME#, TRDY#, IRDY#, STOP#, LOCK#, PAR, SERR#, DEVSEL# Output Enable Delay from PCLKIN Rising	2.0		ns		
t50c	C/BE[3:0]#, FRAME#, TRDY#, IRDY#, STOP#, LOCK#, PAR, SERR#, DEVSEL# Float Delay from PCLKIN Rising		28.0	ns		
t50d	C/BE[3:0]#, FRAME#, TRDY#, IRDY#, STOP#, LOCK#, PAR, SERR#, DEVSEL# Setup Time to PCLKIN Rising	7.0		ns		52
t50e	C/BE[3:0]#, FRAME#, TRDY#, IRDY#, STOP#, LOCK#, PAR, SERR#, DEVSEL# Hold Time from PCLKIN Rising	0.0		ns		52
t51a	AD[31:0] Setup Time to PCLKIN Rising	7.0		ns		52
t51b	AD[31:0] Hold Time to PCLKIN Rising	0.0		ns		52
t51c	AD[31:0] Valid Delay from PCICLK	2.0	11.0	ns		57
t52a	PREQ0, PREQ1/LDEV# Setup Time to PCLKIN	12.0		ns		52
t52b	PREQ0, PREQ1/LDEV# Hold Time to PCLKIN	2.0		ns		52
t52c	PGNT1, PGNT0 Valid Delay from PCLKIN	2.0	12.0	ns		51
t53a	PWRGOOD Setup Time to HCLKIN Rising	7.0		ns		52
t53b	PWRGOOD Hold Time to HCLKIN Rising	2.0		ns		52

6.4.6 PSC/IB LINK INTERFACE TIMING

AC Characteristics (V_{DD} = 5V ±5%, T_{CASE} = 0 to 85 °C)

Symbol	Parameter	Min	Max	Units	Notes	Fig
t60a	CMDV#, SIDLE# Valid Delay from HCLKIN Rising	3.0	13.0	ns		51
t60b	CMDV#, SIDLE# Setup Time to HCLKIN Rising	11.0		ns		52
t60c	CMDV#, SIDLE# Hold Time to HCLKIN Rising	1.0		ns		52
t61a	LREQ Setup Time to HCLKIN Rising	10.0		ns		52
t61b	LREQ Hold Time to HCLKIN Rising	1.0		ns		52
t62	LGNT Valid Delay from HCLKIN Rising	3.0	13.0	ns		51

6.4.7 PCI BUS IDE TIMING

AC Characteristics (V_{DD} = 5V ±5%, T_{CASE} = 0 to 85 °C)

Symbol	Parameter	Min	Max	Units	Notes	Fig
t70	LBIDE#, IDE1CS#, IDE3CS#, DIR, IOR#, IOW# Valid Delay from PCLKIN Rising	2.0	12.0	ns		51
t71a	IORDY Falling Setup Time to HCLKIN Rising	18.0		ns		52
t71a1	IORDY Rising Setup Time to HCLKIN Rising	8.0		ns		52
t71b	IORDY Hold Time to HCLKIN Rising	2.0		ns		52
t72	IDEA[2:0] Valid Delay from PCLKIN Rising	2.0	12.0	ns		51
t73a	IDED[15:0] Valid Delay from PCLKIN Rising	2.0	12.0	ns		51
t73b	IDED[15:0] Setup Time to HCLKIN Rising	8.0		ns		52
t73c	IDED[15:0] Hold Time to HCLKIN Rising	2.0		ns		52

6.4.8 AC TEST LOADS

Table 24. AC Test Loads

Capacitive Load	Pin
0 pF	RAS[3:0]#, CAS[7:0]#, WE#, MA[10:0]
0 pF	HD[31:0], HDP[3:0], A[31:30,26:2], RDY#, BRDY#, BOFF#, AHOLD, EADS#, INV, KEN#, CPURST, HOLD, A20M#, CI3E, CI3O2, COE[1:0]#, CWE[1:0]#, TWE#, TAG[8:0], CMDV#, SIDLE#, LGNT, PGNT1, PGNT0, LBIDE#, IDE1CS#, IDE3CS#, DIR, IOR#, IOW#

6.4.9 MISCELLANEOUS CLOCK TIMINGS

Symbol	Parameter	Min	Max	Units	Notes	Fig
t3a	PCICLKIN to HCLKIN Skew	0	0.5	ns	Measured at 1.5V to 2.5V	55

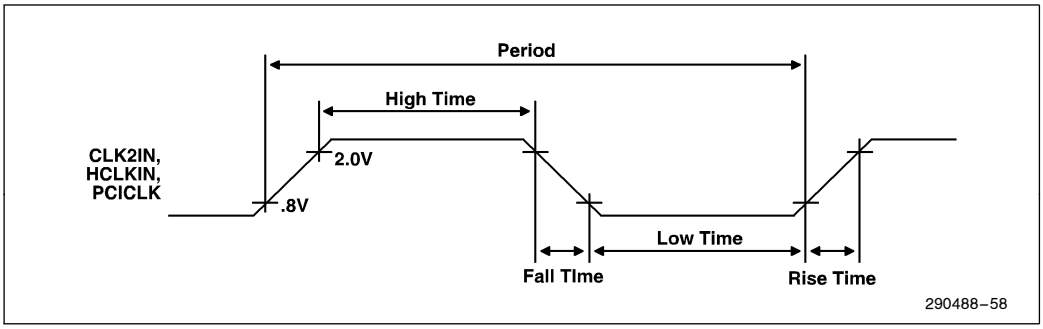


Figure 49. Clock Timing

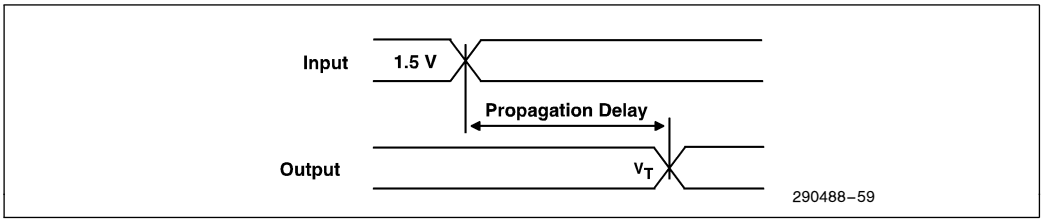


Figure 50. Propagation Delay

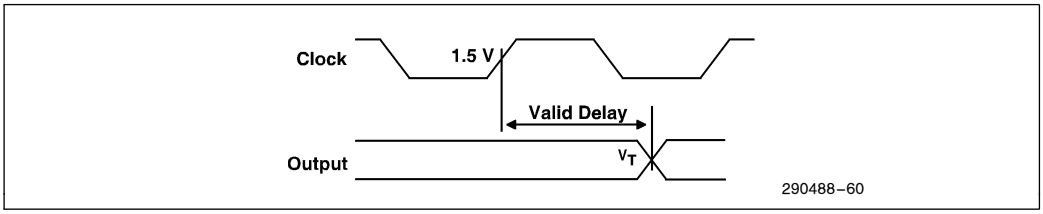


Figure 51. Valid Delay from Rising Clock Edge

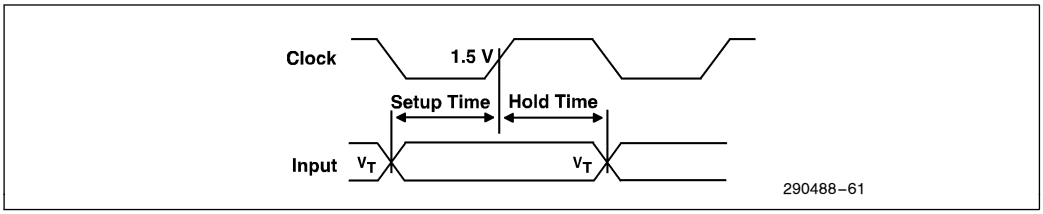


Figure 52. Setup and Hold Times

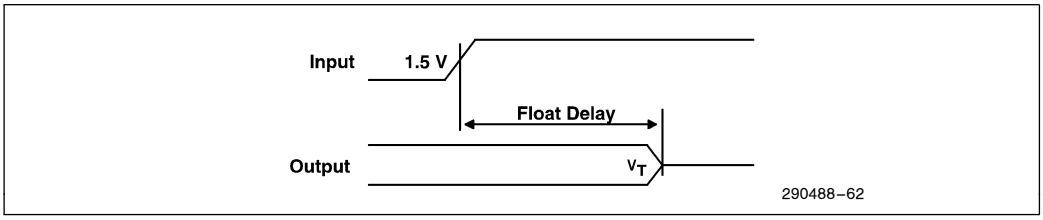


Figure 53. Float Delay

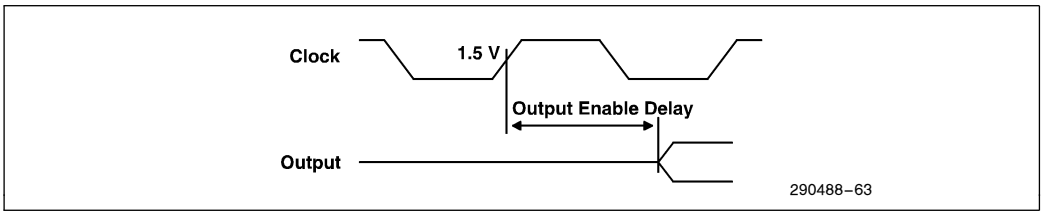
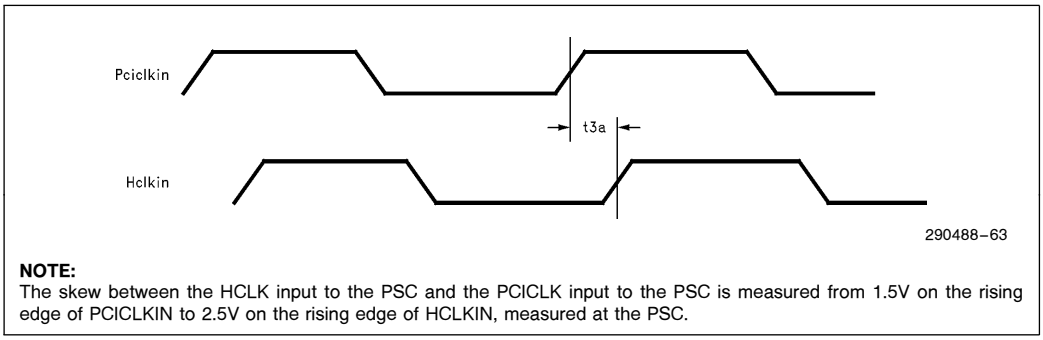


Figure 54. Output Enable Delay



NOTE:
The skew between the HCLK input to the PSC and the PCICLK input to the PSC is measured from 1.5V on the rising edge of PCICLKIN to 2.5V on the rising edge of HCLKIN, measured at the PSC.

Figure 55. PCICLKIN to HCLKIN Skew

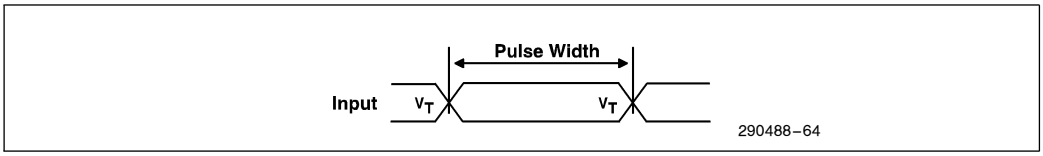


Figure 56. Pulse Width

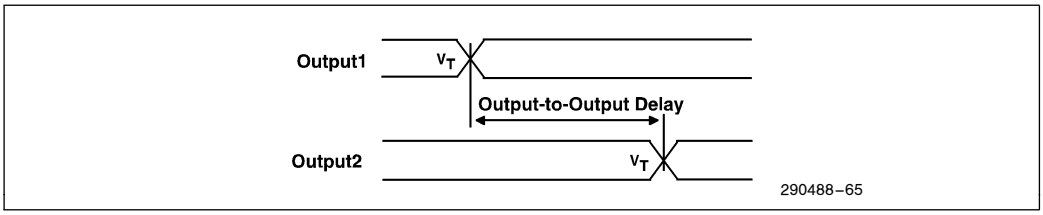


Figure 57. Output-to-Output Delay

7.0 IB AND PSC PACKAGE INFORMATION

Figure 58 shows the package information for the 82426EX IB and Figure 59 shows the package information for the 82425EX PSC.

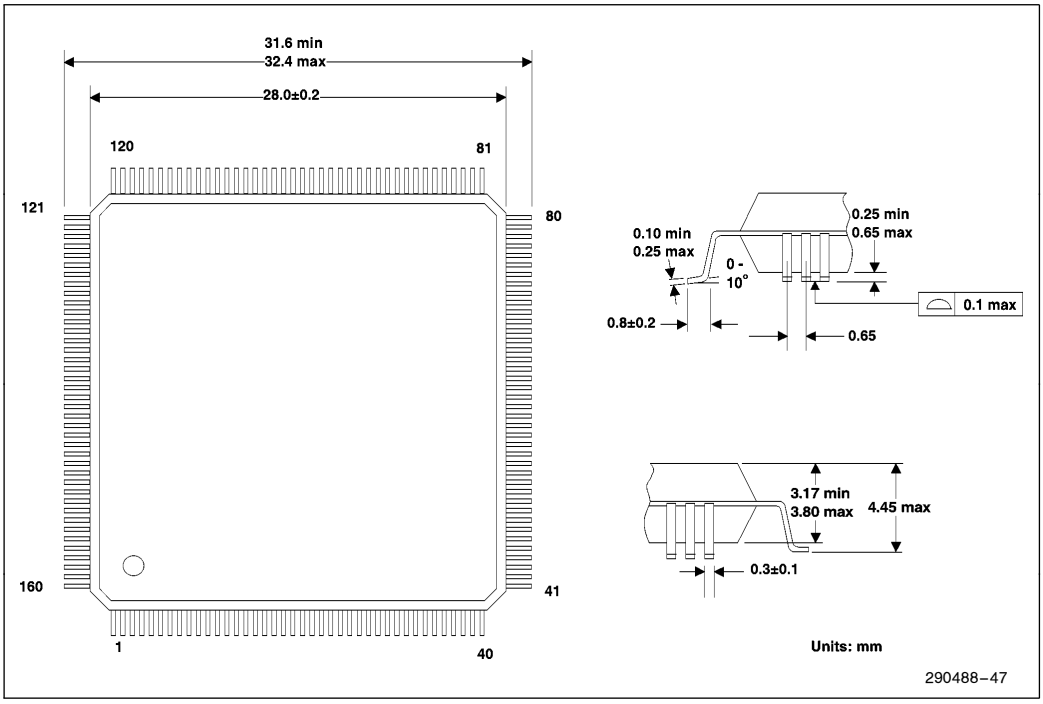


Figure 58. IB Package Dimensions

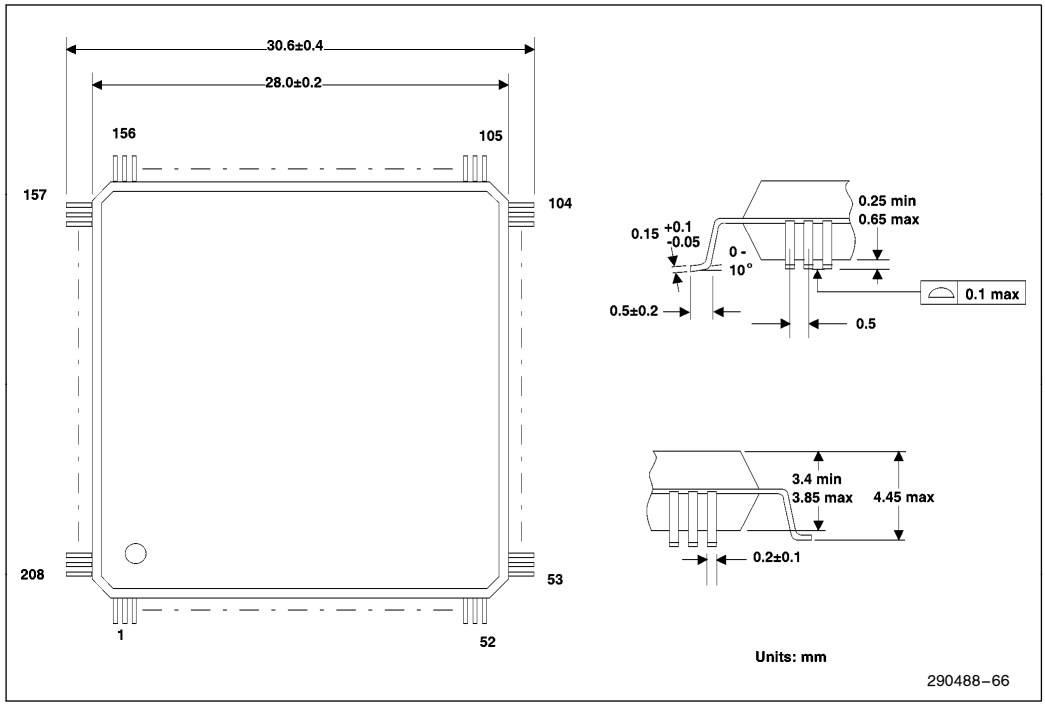


Figure 59. PSC Package Dimensions

7.1 Thermal Characteristics

The 82420EX PCIset is designed for operation at case temperatures between 0°C and 85°C. The thermal resistances of the packages are given in Table 25 and Table 26.

Table 25. PSC Package Thermal Resistance

Parameter	Air Flow (Linear Feet per Minute)		
	0	200	400
θ Junction to Case (°C/Watt)	16	16	16
θ Case to Ambient (°C/Watt)	34	27	23

Table 26. IB Package Thermal Resistance

Parameter	Air Flow (Linear Feet per Minute)		
	0	200	400
θ Junction to Case (°C/Watt)	16	16	16
θ Case to Ambient (°C/Watt)	34	27	23

8.0 TESTABILITY

8.1 PSC Testability

8.1.1 PSC TRI-STATE CONTROL

The PSC can be forced to tri-state all of its output drivers. The LOCK# must be connected to Vcc through a pull-up resistor for normal operation. The PSC will latch the values of LOCK# on the falling edge of CPURST. If these signals have been driven to a logic ‘‘0’’, the PSC will tri-state all of its drivers on the next rising edge of HCLKIN. The PSC will continue to tri-state all drivers until the rising edge of HCKLIN after LOCK# is forced to a logic ‘‘1’’.

8.1.2 PSC NAND TREE

A NAND Tree is provided in the PSC for Automated Test Equipment (ATE) board level testing. The NAND tree allows the tester to set the connectivity of each of the PSC signal pins. While in NAND tree mode, all PSC drivers except A30 are tri-stated. The NAND tree output is driven on pin A30.

NAND tree mode is entered similar to tri-state mode. During CPURST, PREQ0#, like LOCK#, is driven low. NAND tree mode is entered on the rising edge of HCLKIN after CPURST goes inactive.

Table 27 shows the sequence of the NAND tree in the PSC. Non-inverting inputs are driven directly into the input of a NAND gate in the NAND tree.

Table 27. PSC NAND Tree Structure

Tree Output #	Pin #	Pin Name	Type	Comments
	111	A30	NI	A30 is the test mode output
1	39	HCLKIN	NI	End of NAND tree, goes to A30 in test mode
2	208	AD0	NI	
3	207	AD1	NI	
4	206	AD2	NI	
5	205	AD3	NI	
6	204	AD4	NI	
7	203	AD5	NI	
8	202	AD6	NI	
9	199	AD7	NI	
10	198	C/BE0#	NI	
11	197	AD8	NI	
12	196	AD9	NI	
13	195	AD10	NI	
14	194	AD11	NI	
15	192	AD12	NI	
16	191	AD13	NI	
17	190	AD14	NI	
18	189	AD15	NI	
19	188	C/BE1#	NI	
20	186	PAR	NI	
21	184	STOP#	NI	
22	183	DEVSEL#	NI	
23	182	KBDRST#	INV	
24	181	TRDY#	NI	
25	180	IRDY#	NI	
26	179	FRAME#	NI	
27	177	C/BE2#	NI	
28	176	AD16	NI	

Table 27. PSC NAND Tree Structure (Continued)

Tree Output #	Pin #	Pin Name	Type	Comments
29	175	AD17	NI	
30	174	AD18	NI	
31	173	AD19	NI	
32	171	AD20	NI	
33	170	AD21	NI	
34	169	AD22	NI	
35	168	AD23	NI	
36	167	C/BE3#	NI	
37	166	AD24	NI	
38	163	AD25	NI	
39	162	AD26	NI	
40	161	AD27	NI	
41	160	AD28	NI	
42	159	AD29	NI	
43	158	AD30	NI	
44	157	AD31	NI	
45	156	PGNT0#	NI	
46	154	PGNT1#/HRDY#	NI	
47	152	PREQ1#/HDEV#	NI	
48	151	WE#	NI	
49	150	CAS7#	NI	
50	149	CAS3#	INV	
51	148	CAS3#	INV	
52	146	CAS2#	NI	
53	145	CAS5#	NI	
54	144	CAS1#	NI	
55	143	CAS4#	NI	
56	141	CAS0#	NI	

Table 27. PSC NAND Tree Structure (Continued)

Tree Output #	Pin #	Pin Name	Type	Comments
57	140	RAS4#	NI	
58	139	RAS3#	NI	
59	138	RAS2#	NI	
60	137	RAS1#	NI	
61	136	RAS0#	NI	
62	135	COE0#	NI	
63	134	COE1#	NI	
64	133	CI3O2	NI	
65	132	CI3E	NI	
66	131	SMIACT#	INV	
67	130	LBIDE#	NI	
68	129	MA10	NI	
69	127	MA9	NI	
70	126	MA8	NI	
71	125	MA7	NI	
72	124	MA6	NI	
73	123	MA5	NI	
74	122	MA4	NI	
75	121	MA3	NI	
76	120	MA2	NI	
77	118	MA1	NI	
78	117	MA0	NI	
79	116	CLK2IN	NI	
80	115	CWE1#	NI	
81	113	CWEO#	NI	
82	112	AHOLD	NI	
83	110	HDP3	NI	
84	109	HD23	NI	

Table 27. PSC NAND Tree Structure (Continued)

Tree Output #	Pin #	Pin Name	Type	Comments
85	108	HD20	NI	
86	107	HD22	NI	
87	106	HD19	NI	
88	105	HD21	NI	
89	104	HD11	NI	
90	103	HD18	NI	
91	102	HD9	NI	
92	101	HD13	NI	
93	100	HD17	NI	
94	99	HD10	NI	
95	98	HDP1	NI	
96	95	HD8	NI	
97	94	HD15	NI	
98	93	HD12	NI	
99	92	HD24	NI	
100	91	HD25	NI	
101	90	HD3	NI	
102	89	HDP2	NI	
103	88	HD27	NI	
104	87	HD5	NI	
105	86	HD16	NI	
106	85	HD26	NI	
107	83	HD14	NI	
108	82	HD29	NI	
109	81	HD31	NI	
110	80	HD6	NI	
111	79	HD7	NI	
112	78	HD28	NI	

Table 27. PSC NAND Tree Structure (Continued)

Tree Output #	Pin #	Pin Name	Type	Comments
113	77	HD4	NI	
114	76	HD30	INV	
115	75	HD2	NI	
116	74	HD1	NI	
117	73	HDP0	NI	
118	72	HD0	NI	
119	71	EADS#	NI	
120	70	HOLD	NI	
121	69	RDY#	NI	
122	68	KEN#	NI	
123	67	BRDY#	NI	
124	66	PCD/CACHE#	NI	
125	65	D/C#	NI	
126	64	W/R#	NI	
127	63	MI/O#	NI	
128	62	HLDA	NI	
129	59	BLAST#	INV	
130	58	ADS#	INV	
131	57	HITM#	NI	
132	56	BE3#	NI	
133	55	BE2#	NI	
134	54	BE1#	NI	
135	53	BE0#	NI	
136	52	TWE#	NI	
137	51	TAG7	NI	
138	50	TAG6	NI	
139	49	TAG5	NI	
140	48	TAG4	NI	

Table 27. PSC NAND Tree Structure (Continued)

Tree Output #	Pin #	Pin Name	Type	Comments
141	47	TAG3	NI	
142	46	TAG2	NI	
143	45	TAG1	NI	
144	44	TAG0	NI	
145	42	TAG8	NI	
146	41	SRESET/INIT	NI	
147	37	SMI#	INV	
148	36	A4	NI	
149	35	A6	NI	
150	34	A3	NI	
151	33	A2	NI	
152	32	A10	NI	
153	31	A8	NI	
154	30	A7	NI	
155	29	A11	NI	
156	28	A5	NI	
157	26	A9	INV	
158	25	A13	NI	
159	24	A16	NI	
160	23	A20	NI	
161	22	A12	NI	
162	21	A15	NI	
163	20	A22	NI	
164	19	A24	NI	
165	18	A14	NI	
166	17	A18	NI	
167	16	A21	NI	
168	14	A19	NI	

Table 27. PSC NAND Tree Structure (Continued)

Tree Output #	Pin #	Pin Name	Type	Comments
169	13	A23	NI	
170	12	A17	NI	
171	11	A26	NI	
172	9	A25	NI	
173	8	A31	NI	
174	7	PCICLKIN	NI	
175	6	LGNT #	NI	
176	5	LREQ #	NI	
177	4	SIDLE #	NI	
178	3	CMDV #	NI	
179	2	SERR #	NI	Cell furthest from NAND Tree output

NOTES:
NI = Non-Inverting
INV = Inverting

8.1.3 PSC NAND TREE DIAGRAM

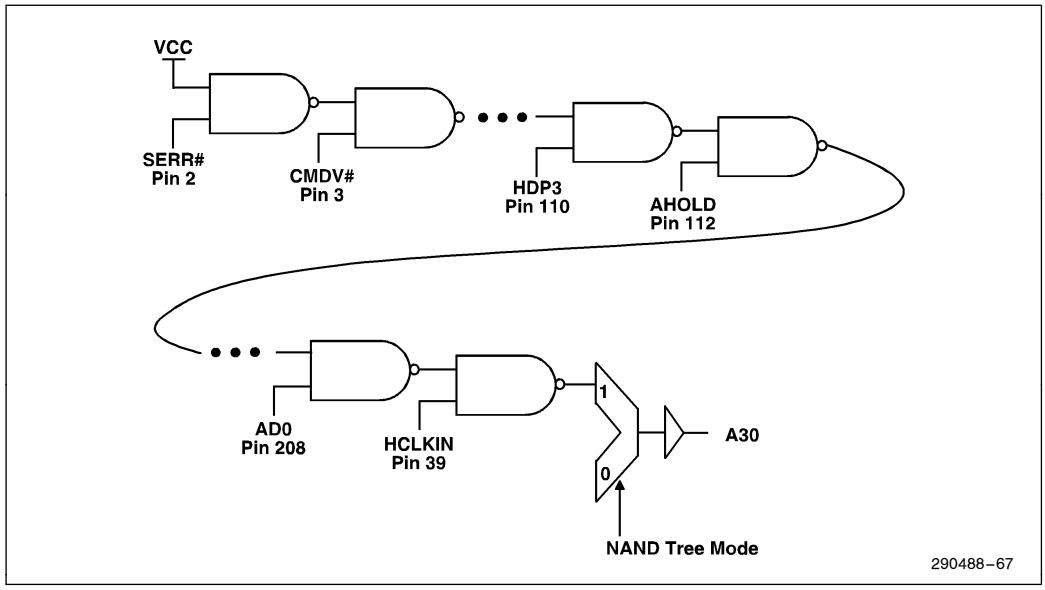


Figure 60. PSC NAND Tree Diagram

8.2 IB Testability

The TESTIN# pin is used to test the IB. During normal operations, the TESTIN# pin must be pulled high through an external pull-up.

8.2.1 IB TRI-STATE

The TESTIN# pin and IRQ3 are used to provide a high impedence tri-state test mode. When the following input combination occurs, all outputs and bidirectional pins are tri-stated, including SPKR:

TESTIN# = 0
IRQ3 = 1
IRQ5 = 0
IRQ6 = 1

The IB must be reset after the bidirectional and output pins have been tri-stated in this manner.

8.2.2 IB NAND TREE

A NAND Tree is provided primarily for VIL/VIH testing. The NAND Tree is also useful for ATE at board level testing. The NAND Tree allows for the tester to test the solder connections for each individual signal pin.

The TESTIN# pin along with IRQ5 and IRQ6 activate the NAND Tree. All outputs and bidirectional pins, except SPKR, are tri-stated when the following input combinations occur:

TESTIN# = 0 and IRQ5 = 1
or
TESTIN# = 0 and IRQ6 = 0

The output pulse train is observed at the SPKR test output, which is not tri-stated while in NAND Tree mode.

The sequence of the ATE test is as follows:

1. Drive TESTIN# low.
2. Drive each input and bidirectional pin noted in Section 8.2.3 high, except for SPKR.
3. Starting with pin 1, SYSCLK, individually drive each pin low. Expect SPKR to toggle with each pin.
4. Turn off tester drivers before driving TESTIN# high.
5. Reset the IB prior to proceeding with further testing.

8.2.3 IB NAND TREE CELL ORDER

NAND Tree cell order is dependent on pin placement. The IB NAND Tree follows pin order around the part from pin 1 to pin 158.

Table 28. IB NAND Tree Structure

Tree Output #	Pin #	Pin Name	Type	Comments
	69	SPKR	NI	Test Mode Output
1	158	SA12	NI	End of NAND Tree, goes to SPKR in test mode
2	157	REFRESH#	NI	
3	156	SA13	NI	
4	155	DREQ1	NI	
5	154	SA14	NI	
6	153	DACK1#	NI	
7	152	SA15	NI	
8	151	DREQ3	NI	
9	149	SA16	NI	
10	148	DACK3#	NI	
11	147	SA17	NI	
12	146	IOR#	NI	
13	145	SA18	NI	
14	144	IOW#	NI	
15	143	SA19	NI	
16	142	SMEMR#	NI	
17	139	AEN	NI	
18	138	SMEMW#	NI	
19	137	IOCHRDY	NI	
20	136	SD0	NI	
21	135	SD1	NI	
22	134	ZEROWS#	NI	
23	133	SD2	NI	
24	132	SD3	NI	
25	131	DREQ2	NI	
26	129	SD4	NI	
27	128	SD5	NI	
28	127	IRQ9	NI	

Table 28. IB NAND Tree Structure (Continued)

Tree Output #	Pin #	Pin Name	Type	Comments
29	126	SD6	NI	
30	125	SD7	NI	
31	124	RSTDRV	NI	
32	123	IOCHK#	NI	
33	122	KBCCS#	NI	
34	119	IRQ1	NI	
35	118	PIRQ0#	NI	
36	117	PIRQ1#	NI	
37	116	SERR#	NI	
	115	TESTIN#	NI	Not part of NAND tree
38	114	PWROK	NI	
39	113	PCICLK2	NI	
40	112	PCICLK1	NI	
41	111	PCIRST#	NI	
42	109	CLK2OUT	NI	
43	108	CLK2IN	NI	
44	107	CMDV#	NI	
45	106	SIDLE#	NI	
46	105	LREQ#	NI	
47	104	LGNT#	NI	
48	103	A17	NI	
49	102	A14	NI	
50	101	A15	NI	
51	99	OSC	NI	
52	98	A12	NI	
53	97	A16	NI	
54	96	A13	NI	
55	95	A9	NI	
56	94	A5	NI	

Table 28. IB NAND Tree Structure (Continued)

Tree Output #	Pin #	Pin Name	Type	Comments
57	93	A11	NI	
58	92	A7	NI	
59	91	A8	NI	
60	89	A10	NI	
61	88	A6	NI	
62	87	A4	NI	
63	86	A2	NI	
64	85	A3	NI	
65	84	SMI#	NI	
66	83	HCLKIN	NI	
67	82	HCLKOUT1	NI	
68	81	HCLKOUT2	NI	
69	79	CPURST	NI	
70	78	SRESET	NI	
71	77	NMI	NI	
72	76	IGNNE#	NI	
73	75	FERR#	NI	
74	73	INTR	NI	
75	72	STPCLK#	NI	
76	71	EXTSMI#	NI	
77	68	RTCALE	NI	
78	67	RTCCS#	NI	
79	66	BIOSCS#	NI	
80	65	XBUSOE#	NI	
81	64	XBUSTR#	NI	
82	63	IRQ8#	NI	
83	62	SD15	NI	
84	61	SD14	NI	

Table 28. IB NAND Tree Structure (Continued)

Tree Output #	Pin #	Pin Name	Type	Comments
85	59	SD13	NI	
86	58	SD12	NI	
87	57	DREQ7	NI	
88	56	SD11	NI	
89	55	DACK7#	NI	
90	54	SD10	NI	
91	53	DREQ6	NI	
92	52	SD9	NI	
93	51	DACK6#	NI	
94	49	SD8	NI	
95	48	DREQ5	NI	
96	47	MEMW#	NI	
97	45	DACK5#	NI	
98	44	MEMR#	NI	
99	43	DREQ0	NI	
100	42	LA17	NI	
101	41	DACK0#	NI	
102	40	LA18	NI	
103	39	IRQ14	NI	
104	37	LA19	NI	
105	36	IRQ15	NI	
106	35	LA20	NI	
107	34	IRQ12/M	NI	
108	33	LA21	NI	
109	32	IRQ11	NI	
110	31	LA22	NI	
111	30	IRQ10	NI	
112	29	LA23	NI	

Table 28. IB NAND Tree Structure (Continued)

Tree Output #	Pin #	Pin Name	Type	Comments
113	28	IOCS16#	NI	
114	26	SBHE#	NI	
115	25	MEMCS16#	NI	
116	24	SA0	NI	
117	23	SA1	NI	
118	22	SA2	NI	
119	21	SA3	NI	
120	18	BALE	NI	
121	17	SA4	NI	
122	16	TC	NI	
123	15	SA5	NI	
124	14	DACK2#	NI	
125	13	SA6	NI	
126	12	IRQ3	NI	
127	11	SA7	NI	
128	9	IRQ4	NI	
129	8	SA8	NI	
130	7	IRQ5	NI	
131	6	SA9	NI	
132	5	IRQ6	NI	
133	4	SA10	NI	
134	3	IRQ7	NI	
135	2	SA11	NI	
136	1	SYSCLK	NI	Cell furthest from NAND Tree output

8.2.4 IB NAND TREE DIAGRAM

Figure 61 shows the NAND Tree diagram. The only function pin not included in the pin order is SPKR, which is used as the Test Output at the end of the tree.

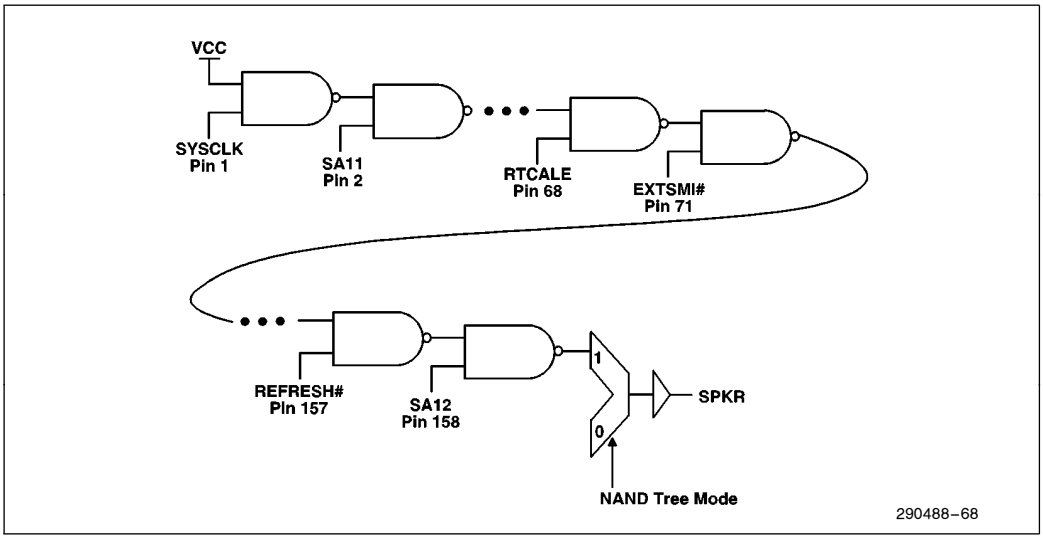


Figure 61. IB NAND Tree Diagram

9.0 REVISION HISTORY

Revision -003 of the 82420EX PCIset data sheet contains updates and improvements to the original version. A revision summary of changes is listed below.

The sections significantly revised since revision -001 are:

Global	The 82420EX PCIset supports host bus operations of 25 and 33 MHz. All references to 50 MHz support have been removed.
Sections 1.1, 1.2, 2.1.1	PGNT1#/HRDY# and M/IO# signal type was incorrectly published. M/IO# and PGNT1#/HRDY# are I/O pins.
	EXTSMI# and PWROK signal type was incorrectly published. EXTSMI# and PWROK are IS pins.
	SERR# was added to Table 3.
Section 2.2.4	Pin description for the OSC pin has been added.
Section 3.1	Data returned during accesses to a reserved register location was incorrectly published. Reads return all 0's during accesses to a reserved register location.
Section 3.3.5	More information is provided on the contents of the Revision Identification Register.
Section 3.3.6	The byte merging feature description has been updated.
Section 3.3.10	Part Revision Register description has been updated.
Section 3.3.20	A warning has been added concerning O/S which size system memory directly without the use of system BIOS.

Section 3.3.21	Bits[2:0] = 000 and bits[2:0] = 001 are now reserved configurations in the SMRAM Control Register.
Section 3.3.22	Changes have been made to the Fast Off Timer Count Granularity.
Section 3.3.25	Changes have been made to the Fast Off Timer count down value.
Section 3.3.26	Bits[15:8] were incorrectly omitted from the SMI Request Register. Bits[15:8] are shown as reserved.
Section 3.3.27	Value of the duration of the STPCLK# asserted period has changed.
Section 3.3.28	Value of the duration of the STPCLK# negated period has changed.
Section 4.3.4	Further information has been added to the keyboard controller circuit description.
Section 4.12.1	Table 19 has been updated. New notes have been added to Table 19.
Section 4.13.6	Table 21 has been updated.
Section 5.0	Section 5.0 has been removed. The 82420EX PCIset electrical characteristics will be published as a separate document. See your Intel representative for a copy of the document with the 82420EX PCIset electrical characteristics.
Section 6.0	Section 6.0 is now Section 5.0. Figure 58 and Figure 59 have changed.
Section 6.1	Section 6.1 is now Section 5.1. Table 25 and Table 26 have changed.

The sections significantly revised since revision -002 are:

Section 3.3.22	The description of bit 1 in the SMI Control Register has changed. Software can only set bit 1 to a 0 by writing a 0 to it.
Section 5.0	Section 5.0 has been added. This is a new section titled Design Considerations.
Section 6.0	Section 6.0 has been added. This section includes the AC, DC and mechanical specifications and timings. The following IB specifications have changed since the electrical characteristics were last published in revision -001: t1c (min), t1f (min). The following PSC specifications have changed since the electrical characteristics were last published in revision -001: t1c (min), t1d (min), t41b (min), t41c (min).

The sections significantly revised since revision -003 are:

1. References to programmable access timings to support 50 MHz operation of the PSC are removed.
2. Section 4.15.1, Clock, Layout/Loading Recommendation, is added.
3. Section 6.3.4, ISA Bus and X-Bus Timings: t6a, t6d, t6e, t16a, t27a, are changed.
4. Section 6.4.9, Miscellaneous Clock Timings, is added.